UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de tecnologías y servicios de telecomunicación

TRABAJO FIN DE GRADO

DESARROLLO DE UN SISTEMA DE RECONSTRUCCIÓN 3D A PARTIR DE IMÁGENES CAPTURADAS POR DRONES

Autor: Ignacio Alberto Ramos Howell
Tutor: Pablo Carballeira López
Ponente: José María Martínez Sánchez

Septiembre 2019

DESARROLLO DE UN SISTEMA DE RECONSTRUCCIÓN 3D A PARTIR DE IMÁGENES CAPTURADAS POR DRONES

AUTOR: Ignacio Alberto Ramos Howell

TUTOR: Pablo Carballeira López

PONENTE: José María Martínez Sánchez

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
September 2019

Resumen

Durante los últimos años, el terreno de la reconstrucción 3D ha avanzado notablemente hasta el punto de acercar esta tecnología a las manos de cualquier persona interesada. Sin embargo, las soluciones para reconstrucción 3D de alto nivel requieren una infraestructura que supone una barrera de entrada muy alta para cualquier usuario.

Este trabajo es una primera aproximación a una solución de reconstrucción 3D de alta calidad y al mínimo coste posible. Para ello se han combinado los avances en la tecnología de drones, que permiten grabación de vídeos de muy alta calidad y estabilidad, junto a redes neuronales convolucionales profundas (DCNN) para segmentación semántica, para intentar emular las estructuras de captura de imágenes para reconstrucción 3D de alto nivel.

El primer paso en este proceso ha sido el reconocimiento del objeto a reconstruir en imágenes muestreadas de un video grabado por dron mediante el método de segmentación DeepLabv2 entrenado con COCO-Stuff y PASCAL VOC 12, los cuales han dado buenos resultados para objetos grandes y comunes como personas y coches. Tras comprobar el rendimiento de la segmentación semántica, se ha automatizado el proceso de obtención de máscaras para el set de imágenes.

A continuación, se ha automatizado el proceso de obtención de modelo 3D a partir de videos obtenidos por drones incorporando la obtención de máscaras desarrollado en el primer punto. Este proceso tiene tres partes diferenciadas: (i) lectura del video y muestreado de imágenes, (ii) obtención de máscaras de las imágenes extraídas y (iii) flujo de trabajo de reconstrucción del software de reconstrucción 3D de terceros Agisoft PhotoScan.

Finalmente, se han realizado pruebas para refinar y mejorar los distintos pasos del proceso hasta llegar a tener un proceso de reconstrucción 3D automatizado con una calidad suficiente para esta primera aproximación, analizando qué limitaciones pueden ser evitadas en el futuro para mejorar estos resultados.

Palabras clave

Video, 3D, Reconstrucción 3D, Fotogrametría, Segmentación semántica, Drones, UAV.

Abstract

Over the last years, the 3D reconstruction landscape has advanced significantly to the point of bringing this technology to the hands of any person interested. However, high-level 3D reconstruction solutions require an infrastructure that is a very high barrier to entry for any user.

This work is a first approach to a 3D reconstruction solution of high quality and at the lowest possible cost. To this end, advances in drone technology, which allow recording of videos of very high quality and stability, have been paired together with deep convolutional neural networks (DCNN) for semantic segmentation to try to emulate image capture structures for high-level 3D reconstruction.

The first step in this process has been the recognition of the object to be reconstructed in images sampled from a video recorded by a drone using the DeepLabv2 segmentation method trained with COCO-Stuff and PASCAL VOC 12, which have given good results for large and common objects such as people and cars. After checking the performance of this semantic segmentation method, the process of obtaining masks for the set of images has been automated.

Next, the process of obtaining a 3D model from videos obtained by drones has been automated, incorporating the mask obtention process developed in the first step. This process has three distinct parts: (i) video reading and image sampling, (ii) obtaining masks of the extracted images and (iii) 3D reconstruction workflow of the third-party 3D reconstruction software Agisoft PhotoScan.

Finally, tests have been carried out to refine and improve the different steps of the process until we felt we had developed an automated 3D reconstruction process with sufficient quality for this first approach, analyzing what limitations can be avoided in the future to improve these results.

Keywords

Video, 3D, 3D Reconstruction, Photogrammetry, Semantic segmentation, Drones, UAV.

Agradecimientos

A mi padre, cuyo esfuerzo a lo largo de su vida me ha permitido tener una situación privilegiada.

A mi madre, que siempre está conmigo.

A mi hermano Francisco, que cree más en mi de lo que merezco.

A mi "familia", los amigos de toda la vida que han estado ahí en todo momento y han sido mi mayor apoyo.

A mis amigos de la universidad que me han acompañado y me han ayudado todos estos años: Manu, Pali, Erik, Paula, Gabriel, Edu, etc.

A Diego y Lorenzo, que me han servido más café a mí que a todo el resto de la escuela juntos.

A todas esas personas que estos años han ocupado un sitio especial y me han enseñado mucho sobre la vida y sobre mí (IBN)

Y, por último, a Idoia, pues sin su esfuerzo y preocupación no estaría escribiendo estas palabras.

INDICE DE CONTENIDOS

1 Introducción 9
1.1 Motivación 9
1.2 Objetivos 10
1.3 Organización de la memoria 10
2 Estado del arte 11
2.1 Sistemas para obtención de modelos 3D 11
2.1.1 Sistemas de bajo presupuesto para aficionados 11
2.1.2 Sistemas de arrays de cámaras de uso profesional 12
2.1.3 Captura de imágenes con drones 12
2.1.4 Nuestro planteamiento 13
2.2 Segmentación semántica 13
2.2.1 Datasets y métricas 14
2.2.2 Modelos de segmentación 15
3 Diseño y desarrollo 18
3.1 Proceso completo de reconstrucción 3D a partir de video 18
3.2 Captura de video con dron 18
3.3 Muestreo de imágenes y obtención de máscaras 18
3.3.1 Selección de objeto a segmentar 18
3.3.2 Muestreo de imágenes 20
3.3.3 Obtención de máscaras 20
3.4 Reconstrucción 3D 21
3.4.1 Alineamiento de puntos 21
3.4.2 Construcción de nube densa 23
3.4.3 Generación de la malla 23
3.4.4 Creación de textura 24
4 Integración, pruebas y resultados 25
4.1 Secuencias utilizadas 25
4.2 Segmentación 25
4.2.1 Pruebas con los modelos entrenados 26
4.2.2 Resultados de reconstrucción 3D 27
4.2.3 Conclusiones 28
4.3 Posprocesado de máscaras 28
4.3.1 Problemas observados 29
4.3.2 Operaciones realizadas 30
4.3.3 Resultados del posprocesado de máscaras 30
4.3.4 Conclusiones 32
4.4 Número de imágenes muestreadas 33
4.4.1 Método de valoración 33
4.4.2 Resultados de reconstrucción 3D 33
4.4.3 Conclusiones 34
4.5 Proceso completo de reconstrucción 3D desde el vídeo 34
4.5.1 Resultados de reconstrucción 3D 35
4.5.2 Evaluación del resultado final 36
5 Conclusiones y trabajo futuro 37
5.1 Conclusiones 37
5.1.1 Trabajo realizado 37
5.1.2 Resultados obtenidos 37
5.1.3 Limitaciones observadas 38

5.2 Trabajo futuro 39
5.2.1 Optimización de la captura mediante programación de ruta de vuelo de dron 39
5.2.2 Refinamiento y optimización del proceso de reconstrucción 3D 39
Referencias ..... - 1 -
Anexos ..... - 4 -
Anexo 1. Características de la cámara de Mavic Air ..... - 4 -
Anexo 2. Etiquetas de datasets ..... - 5 -
COCO-Stuff ..... - 5 -
VOC12 ..... - 5 -
Glosario ..... - 7 -

INDICE DE FIGURAS

FIGURA 2.1. EJEMPLOS DE ESCÁNERES 3D DIY. (A) THE $30 3D SCANNER V7. (B) DIY STANDALONE 3D SCANNER POR JUN TAKEDA. (C) OPENSCAN VERSIÓN INICIAL. (D) OPENSCAN VERSIÓN FINAL.....11

FIGURA 2.2. ESTRUCTURAS PARA CAPTURA DE IMÁGENES PARA FOTOGRAMETRÍAS. (A) PI3DSCAN. FUENTE: HTTP://WWW.PI3DSCAN.COM (B) ESTRUCTURA DE CÁMARAS DSLR. FUENTE: HTTPS://PIXELLIGHTEFFECTS.COM.....12

FIGURA 2.3. EJEMPLO DEL DATASET PASCAL VOC 2012 PARA SEGMENTACIÓN DE IMÁGENES. FUENTE: HTTP://HOST.ROBOTS.OX.AC.UK/PASCAL/VOC/VOC2012/INDEX.HTML.....14

FIGURA 2.4. IMÁGENES DEL DATASET COCO-STUFF CON ANOTACIONES A NIVEL DE PIXEL DENSAS PARA *STUFF* Y *THINGS*. FUENTE: [5].....15

FIGURA 2.5. ESQUEMA DE DEEPLABV2. FUENTE: **[**1**]**.....17

FIGURA 2.6. PUNTUACIONES DE LOS DISTINTOS MODELOS PRESENTADOS EN EL RETO PASCAL VOC 2012. 17

FIGURA 3.1. PROCESO COMPLETO DE RECONSTRUCCIÓN 3D A PARTIR DE SECUENCIA GRABADA CON DRON. 18

FIGURA 3.2. FLUJO DE TRABAJO BÁSICO DE PYTORCH. FUENTE: HTTPS://WWW.LEARNOPENCV.COM 18

FIGURA 3.3. PROCESO DE SEGMENTACIÓN SEMÁNTICA REALIZADO POR DEEPLABV2. LAS SEÑALES SE MUESTREAN CON CONVOLUCIÓN DILATADA DESDE 32X A 8X PARA EXTRAER LAS CARACTERÍSTICAS. POSTERIORMENTE SE INTERPOLA PARA AGRANDAR LOS MAPAS DE CARACTERÍSTICAS A LA RESOLUCIÓN ORIGINAL DE LA IMAGEN. FINALMENTE, SE UTILIZAN CRF COMPLETAMENTE CONECTADOS PARA REFINAR EL RESULTADO DE LA SEGMENTACIÓN Y CAPTURAR LAS FRONTERAS DE LOS OBJETOS DE MEJOR MANERA. 19

FIGURA 3.4. REPRESENTACIÓN DE LAS DISTINTAS CLASES LOCALIZADAS EN LA IMAGEN ANALIZADA CON EL MODELO DE DEEPLABV2 ENTRENADO CON COCO-STUFF 164K. 20

FIGURA 3.5. PROCESO DE DILATACIÓN Y EROSIÓN SOBRE LA MÁSCARA 1 (A) Y LA MÁSCARA 2 (B). LA PRIMERA IMAGEN CORRESPONDE A LA MÁSCARA POST-CRF, LA SEGUNDA A LA MÁSCARA TRAS DILATACIÓN Y LA TERCERA A LA MÁSCARA POST-CIERRE. 21

FIGURA 3.6. SPARSE CLOUD, O NUBE ESCASA/POCO DENSA DE PUNTOS JUNTO A LAS POSICIONES DE CÁMARA ESTIMADAS. SE PUEDE VER ALREDEDOR DEL OBJETO A RECONSTRUIR LA CAJA DELIMITADORA. 22

FIGURA 3.7. (A) RESULTADO DE CREACIÓN DE NUBE DENSA DE PUNTOS EN LA SECUENCIA PERSONA CON LOS PARÁMETROS DE CALIDAD ULTRA ALTO Y DE FILTRADO DE PROFUNDIDAD MODERADO. (B) RESULTADO DE LA GENERACIÓN DE MALLA. (C) MODELO FINAL CON SUPERFICIE TEXTURIZADA. 24

FIGURA 4.1. (A) IMÁGENES DE SECUENCIA PERSONA. (B) IMÁGENES DE SECUENCIA COCHE.....25

FIGURA 4.2. MÁSCARAS OBTENIDAS MANUALMENTE DE SECUENCIA PERSONA (A) Y SECUENCIA COCHE (B) 25

FIGURA 4.3. MÁSCARAS OBTENIDAS CON LOS MODELOS DE SEGMENTACIÓN EVALUADOS SUPERPUESTAS CON LAS IMÁGENES ORIGINALES PARA COMPARACIÓN EN LA SECUENCIA PERSONA. (A) COCO-STUFF 10K PRE-CRF Y (B) POST-CRF. (C) COCO-STUFF 164K PRE-CRF Y (D) POST-CRF. (E) PASCAL VOC 12 PRE-CRF Y (F) POST-CRF 26

FIGURA 4.4. MÁSCARAS OBTENIDAS CON LOS MODELOS DE SEGMENTACIÓN EVALUADOS SUPERPUESTAS CON LAS IMÁGENES ORIGINALES PARA COMPARACIÓN EN LA SECUENCIA COCHE. (A) COCO-STUFF 10K PRE-CRF Y (B) POST-CRF. (C) COCO-STUFF 164K PRE-CRF Y (D) POST-CRF. (E) PASCAL VOC 12 PRE-CRF Y (F) POST-CRF 27

FIGURA 4.5. RESULTADO DE RECONSTRUCCIÓN 3D DE LA SECUENCIA PERSONA UTILIZANDO LAS MÁSCARAS OBTENIDAS CON EL MODELO ENTRENADO CON PASCAL VOC 12. 27

FIGURA 4.6. RESULTADO DE RECONSTRUCCIÓN 3D DE LA SECUENCIA COCHE UTILIZANDO LAS MÁSCARAS OBTENIDAS CON EL MODELO ENTRENADO CON PASCAL VOC 12. 28

FIGURA 4.7. EJEMPLO DE IRREGULARIDADES EN LA TEXTURA EN LA RECONSTRUCCIÓN DE LA SECUENCIA PERSONA. 29

FIGURA 4.8. EJEMPLO DE IRREGULARIDADES EN LA TEXTURA EN LA RECONSTRUCCIÓN DE LA SECUENCIA COCHE. 29

FIGURA 4.9. PROCESO DE DILATACIÓN Y EROSIÓN SOBRE LA MÁSCARA 1 (A) Y LA MÁSCARA 2 (B). LA PRIMERA IMAGEN CORRESPONDE A LA MÁSCARA POST-CRF, LA SEGUNDA A LA MÁSCARA TRAS DILATACIÓN Y LA TERCERA A LA MÁSCARA POST-CIERRE. 30

FIGURA 4.10. MÁSCARAS OBTENIDAS DE LA SECUENCIA PERSONA ANTES DE PROCESAR (A), Y DESPUÉS DE PROCESAR (B) 30

FIGURA 4.11. MÁSCARAS OBTENIDAS DE LA SECUENCIA COCHE ANTES DE PROCESAR (A), Y DESPUÉS DE PROCESAR (B) 31

FIGURA 4.12. COMPARACIÓN DE LA RECONSTRUCCIÓN 3D DE LA SECUENCIA PERSONA PRE Y POST PROCESADO DE LAS MÁSCARAS. (A) RECONSTRUCCIÓN CON MÁSCARAS SIN PROCESAR. (B) RECONSTRUCCIÓN CON MÁSCARAS PROCESADAS. PODEMOS OBSERVAR EN LAS MANGAS COMO EN (B) SE HA CONSEGUIDO UNA LIGERA MEJORA EN LA SUAVIDAD DE LA TEXTURA DE LOS HOMBROS CON RESPECTO A LAS IRREGULARIDADES OBSERVADAS EN (A) 32

FIGURA 4.13. COMPARACIÓN DE LA RECONSTRUCCIÓN 3D DE LA SECUENCIA COCHE PRE Y POST PROCESADO DE LAS MÁSCARAS. (A) RECONSTRUCCIÓN CON MÁSCARAS SIN PROCESAR. (B) RECONSTRUCCIÓN CON MÁSCARAS PROCESADAS. PODEMOS OBSERVAR EN EL LATERAL IZQUIERDO DEL COCHE COMO EN (B), EN ESTE CASO, SE HA CONSEGUIDO UNA NOTABLE MEJORA EN LA SUAVIDAD DE LA TEXTURA CON RESPECTO A LAS IRREGULARIDADES OBSERVADAS EN (A). 32

FIGURA 4.14. ALINEAMIENTO DE CÁMARAS DE LA SECUENCIA PERSONA PARA DISTINTOS NÚMEROS DE IMÁGENES. (A) 24 IMÁGENES. (B) 60 IMÁGENES. (C) 72 IMÁGENES. (D) 120 IMÁGENES. 34

FIGURA 4.15. ALINEAMIENTO DE CÁMARAS DE LA SECUENCIA COCHE PARA DISTINTOS NÚMEROS DE IMÁGENES. (A) 48 IMÁGENES. (B) 60 IMÁGENES. (C) 72 IMÁGENES. (D) 120 IMÁGENES. 34

FIGURA 4.16. RESULTADOS DE RECONSTRUCCIÓN 3D DE LA SECUENCIA PERSONA. (A) RECONSTRUCCIÓN AUTOMÁTICA SIN USO DE MÁSCARAS. (B) RECONSTRUCCIÓN MANUAL SIN USO DE MÁSCARAS Y AJUSTE DE LA CAJA DELIMITADORA ANTES DE CREAR LA NUBE DENSA. (C) RECONSTRUCCIÓN AUTOMÁTICA CON USO DE MÁSCARAS.....35

FIGURA 4.17. RESULTADOS DE RECONSTRUCCIÓN 3D DE LA SECUENCIA COCHE. (A) RECONSTRUCCIÓN AUTOMÁTICA SIN USO DE MÁSCARAS. (B) RECONSTRUCCIÓN MANUAL SIN USO DE MÁSCARAS Y AJUSTE DE LA CAJA DELIMITADORA ANTES DE CREAR LA NUBE DENSA. (C) RECONSTRUCCIÓN AUTOMÁTICA CON USO DE MÁSCARAS.....36

FIGURA 5.1. EJEMPLO DEL EFECTO DE SOLO TENER EL PUNTO DE VISTA SUPERIOR Y COMO AFECTA A LA RECONSTRUCCIÓN. (A) VISTA SUPERIOR DE LA SECUENCIA RECONSTRUIDA. (B) VISTA INFERIOR DE LA MISMA ZONA DONDE SE PUEDE VER EL EFECTO DE NO TENER IMÁGENES CAPTADAS DESDE OTRAS ALTURAS. 38

FIGURA 5.2. EJEMPLO DE RUTAS PROGRAMABLES PARA LA CAPTURA DE SECUENCIAS PARA RECONSTRUCCIÓN 3D. RUTA EN ESPIRAL (A). RUTA EN ZIG-ZAG (B).....39

INDICE DE TABLAS

TABLA 3.1. CLASES LOCALIZADAS EN LA IMAGEN DE ENTRADA REPRESENTADA EN LA FIGURA 3.2 Y SUS RESPECTIVAS ETIQUETAS.....20

TABLA 3.2. IMÁGENES UTILIZADAS PARA CADA DISTINTO VALOR DEL PARÁMETRO DE PRECISIÓN DE ALINEAMIENTO DE FOTOS. 22

TABLA 3.3. IMÁGENES UTILIZADAS PARA CADA DISTINTO VALOR DEL PARÁMETRO DE CALIDAD EN LA CREACIÓN DE NUBE DENSA DE PUNTOS.....23

TABLA 3.4. COMPARACIÓN DE LOS PARÁMETROS DE FILTRADO DE PROFUNDIDAD DE AGISOFT PHOTOSCAN. 23

TABLA 4.1. MEDIDAS DE EVALUACIÓN DE LOS MODELOS DE SEGMENTACIÓN UTILIZADOS SOBRE LA SECUENCIA PERSONA.....26

TABLA 4.2. MEDIDAS DE EVALUACIÓN DE LOS MODELOS DE SEGMENTACIÓN UTILIZADOS SOBRE LA SECUENCIA COCHE. 27

TABLA 4.3. MEDIDAS DE PORCENTAJE DE COINCIDENCIA DE MÁSCARAS OBTENIDAS PARA PERSONA CON LAS DE REFERENCIA DE LA FIGURA 4.2. 31

TABLA 4.4. MEDIDAS DE PORCENTAJE DE COINCIDENCIA DE MÁSCARAS OBTENIDAS PARA COCHE CON LAS DE REFERENCIA DE LA FIGURA 4.2. 31

TABLA 4.5. DATOS DE RECONSTRUCCIÓN 3D PARA DISTINTO NÚMERO DE IMÁGENES PARA EL OBJETO PERSONA 3. 33

TABLA 4.6. DATOS DE RECONSTRUCCIÓN 3D PARA DISTINTO NÚMERO DE IMÁGENES PARA EL OBJETO COCHE. 33

TABLA 4.7. MEDIDAS DE PROCESAMIENTO DE RECONSTRUCCIÓN 3D DE LOS DISTINTOS CASOS MOSTRADOS EN LAS FIGURAS 4.16 Y 4.17. 36

1 Introducción

1.1 Motivación

Durante los últimos años ha habido una evolución notable en la tecnología integrada en los drones comerciales [1], dando lugar a la salida al mercado de drones con gran capacidad para capturar video de buena calidad.

Junto a esto, han aparecido soluciones de reconstrucción 3D a partir de fotografías tomadas por drones comerciales como pueden ser Pix4DMapper [17], DroneDeploy [18] y OpenDroneMap [19]. Sin embargo, estas soluciones están más orientadas al mapeo 3D de superficies de terreno y a generación de modelos 3D de estructuras grandes donde solo es posible obtener imágenes mediante drones.

Para la captura de objetos o estructuras de tamaño inferior, siguen por tanto existiendo las mismas técnicas de captura, que van desde la captura manual de imágenes de un objeto con un teléfono móvil que cualquier particular podría realizar, hasta estructuras complejas de arrays de cámaras DSLR con captura automatizada que se usan profesionalmente [30].

En todos estos sistemas, desde los de menor presupuesto hasta los de uso profesional, hemos observado algunas limitaciones en relación a la reconstrucción 3D:

- **Proceso de captura**
    - Para soluciones de bajo presupuesto, el proceso de captura puede ser largo y tedioso al tener que capturar las imágenes de una en una desde distintas posiciones.
- **Tamaño de objetos capturados**
    - Especialmente en los sistemas de bajo presupuesto, el tamaño de los objetos de los que podemos obtener modelos 3D va a ser, en general, pequeño ya que para tener consistencia en la captura de las imágenes estaremos muy limitados por la estructura diseñada para ello. Podemos ver ejemplos en el Apartado 2.1.1.
    - En los sistemas de alto presupuesto, el tamaño de los objetos que podemos capturar aumenta considerablemente, pero sigue estando limitado tanto por espacio, como por inmovilidad de la estructura, como veremos en el Apartado 2.1.2
- **Entorno no controlado**
    - Si queremos hacer captura de objetos más grandes y llevamos la captura al exterior, perdemos el control del entorno, lo cual puede dificultar el proceso de reconstrucción 3D debido a que tanto el fondo como la iluminación pueden contribuir negativamente e impedir una reconstrucción correcta al afectar a la detección de puntos de interés y su pareado entre imágenes.
- **Edición posterior**
    - Incluso en reconstrucciones buenas con entorno no controlado o semicontrolado, muchas veces es necesario un trabajo de limpieza de todos aquellos elementos reconstruidos que no formen parte del objeto de interés.
    - En los sistemas con entorno controlado, suelen introducirse marcadores para realizar un acotamiento de la zona de reconstrucción, pero esto obviamente introduce una limitación en el tamaño del objeto capturado.

Consideramos que todas estas limitaciones se pueden superar aportando un nuevo enfoque en la captura de las imágenes de objetos y estructuras de cualquier tamaño, utilizando drones para grabar secuencias, reduciendo los tiempos de capturas y proporcionando un gran número de imágenes; así como añadiendo un proceso de segmentación del objeto de interés del fondo que creemos contribuirá a la mejora de tiempos y del modelo generado.

1.2 Objetivos

El principal objetivo de este trabajo es desarrollar un sistema de reconstrucción 3D automatizado escalable capaz de emular sistemas de alto presupuesto utilizando drones para capturar las imágenes. Para ello deberemos prestar especial atención al proceso de captura de secuencia y calidad de las imágenes obtenidas, y al efecto de un entorno no controlado y su efecto en la reconstrucción.

Los objetivos de este trabajo serán los siguientes:

1. Aportar un nuevo enfoque al proceso de captura de imágenes para reconstrucción 3D utilizando drones, consiguiendo reducir notablemente el tiempo de captura de las imágenes grabando una secuencia alrededor del objeto y muestreando imágenes.
2. Automatización del proceso de reconstrucción 3D. Nuestro objetivo será desarrollar un único script que ejecute todo el proceso de reconstrucción 3D obteniendo previamente máscaras del objeto de interés de las imágenes mediante la utilización de algoritmos de estado del arte de segmentación semántica.

1.3 Organización de la memoria

La documentación de este trabajo va a seguir la siguiente estructura:

- En el *capítulo 2* se hablará del estado del arte en cada uno de los distintos elementos que vamos a integrar para el diseño del sistema de reconstrucción 3D.
- En el *capítulo 3* se detallará el proceso completo de reconstrucción 3D automatizado desde la captura del video hasta la obtención del modelo.
- El c*apítulo 4* estará dedicado a documentar las pruebas realizadas y analizar los distintos resultados obtenidos.
- Finalmente, en el *capítulo 5* se comentarán las conclusiones a las que se han llegado a lo largo del desarrollo del trabajo y algunas posibles líneas de trabajo futuro.

2 Estado del arte

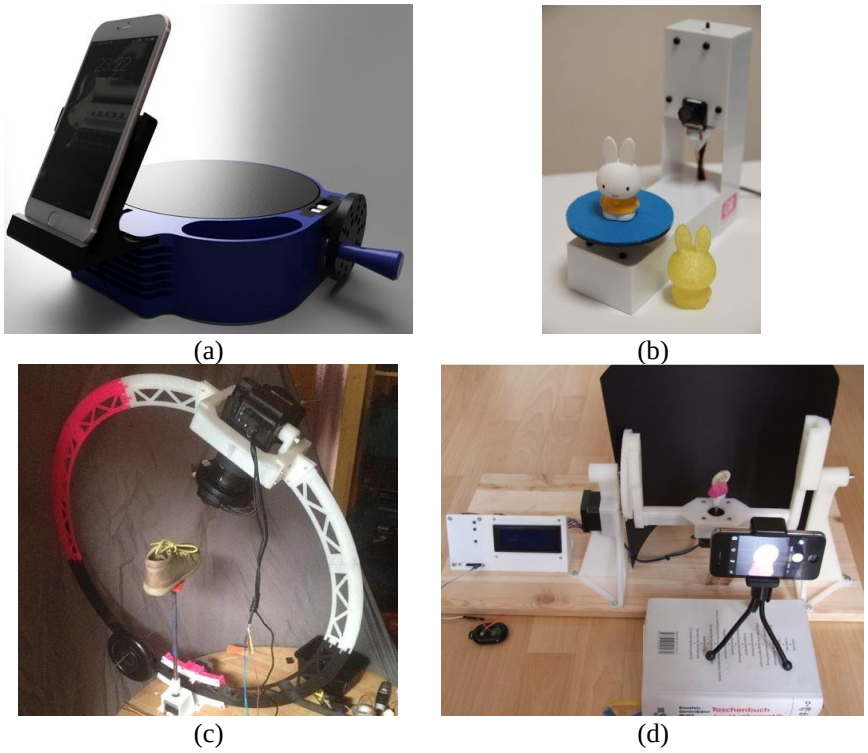
2.1 Sistemas para obtención de modelos 3D

Uno de los objetivos de este trabajo es emular sistemas diseñados para la captura de imágenes destinadas a fotogrametría. A continuación, vamos a presentar los sistemas más comunes y en los que nos hemos basado durante el proceso de diseño de nuestra solución.

2.1.1 Sistemas de bajo presupuesto para aficionados

Habiendo avanzado tanto el software de reconstrucción 3D, y con los avances en el terreno de los smartphones, que prácticamente permiten a cualquier persona llevar una cámara de buena calidad en su bolsillo, muchos entusiastas del 3D han diseñado sus propios sistemas de captura de imágenes para fotogrametría, algunos específicamente para smartphones, otros utilizando cámaras integradas. Entre estos sistemas podemos destacar cuatro diseños comunes que se basan en unos mismos conceptos:

1. Sistema de captura manual con turntable. Figura 2.1a.
2. Sistema de captura automático con turntable. Figura 2.1b.
3. Sistema de captura automática con cámara móvil. Figura 2.1c.
4. Sistema de captura automática con cámara fija. Figura 2.1d.



(a) (b) (c) (d)

Figura 2.1. Ejemplos de escáneres 3D DIY. (a) The $30 3D scanner V7. (b) DIY Standalone 3D Scanner por Jun Takeda. (c) OpenScan versión inicial. (d) OpenScan versión final

Como puede verse en la Figura 2.1, estos sistemas tienen la limitación de solo ser válidos para obtener imágenes de objetos de tamaño reducido.

2.1.2 Sistemas de arrays de cámaras de uso profesional

Los sistemas de captura de imágenes para reconstrucción 3D de uso profesional están formados por estructuras robustas con cientos de cámaras DSLR incorporadas. El sistema que mostramos en la Figura 2.2b a continuación está formado por 144 cámaras DSLR.

Estos sistemas permiten producir reconstrucciones 3D de muy alta calidad, lo que hace que sean utilizados para cualquier producción dentro de la industria VFX.

(a) (b)

Figura 2.2. Estructuras para captura de imágenes para fotogrametrías. (a) Pi3DScan. Fuente: http://www.pi3dscan.com (b) Estructura de cámaras DSLR. Fuente: https://pixellightef-fects.com

Basándose en sistemas de arrays de cámaras DSLR, ha habido proyectos independientes de emular estos sistemas reduciendo el coste al máximo. Este es el caso de Pi3DScan.

Pi3DScan es una estructura de arrays de cámaras, similar a la estructura profesional de cámaras DSLR, que utiliza la popular Raspberry Pi junto con la cámara PI de 5 u 8 megapíxeles para capturas las imágenes. Está compuesto por 100 conjuntos de Raspberry Pi y cámara como podemos ver en la Figura 2.2a.

Estos sistemas tienen como principal limitación la inmovilidad, ya que al ser tan complejos moverlos de localización requiere mucho trabajo de montaje.

2.1.3 Captura de imágenes con drones

Con los avances en la tecnología de drones en los últimos tiempos, entre los que, para los propósitos de este trabajo, podemos destacar la estabilización giroscópica, incorporación de cámaras de alta calidad en estructuras con gimbals y control de inclinación y la posibilidad de planificar vuelos inteligentes [29], se han podido construir drones capaces de obtener imágenes de alta calidad que han sido principalmente aplicadas a la geomática [12]: agricultura, ciencias forestales, arqueología y arquitectura, medio ambiente, gestión de emergencias y monitorización de tráfico. Todas estas aplicaciones utilizan tomas aéreas para realizar reconstrucciones 3D de la superficie terrestre, a excepción de aquellas destinadas a arqueología, que combinan estas tomas aéreas con otras a pie de tierra para obtener modelos 3D de localizaciones de interés.

En los últimos años, sin embargo, los avances mencionados en el párrafo anterior se han aplicado a drones comerciales, como serían las gamas de DJI: Phantom, Mavic y Matrice, o

de Parrot: Bebop y Anafi, que, unido a la aparición de aplicaciones como Pix4D [17], DroneDeploy [18] y OpenDroneMap [19], destinadas a mapeo y modelado 3D con posibilidad de planificar vuelo, ha hecho del modelado 3D con drones una realidad para cualquier particular.

Sin embargo, estos sistemas y softwares para reconstrucción 3D tienen sus limitaciones, de las cuales las que más nos interesa mencionar son las restricciones a la hora de planificar vuelos, como sería la altura, que limita las localizaciones y objetos que podemos capturar; y no estar diseñado para reconstruir objetos aislados, obligando a reconstruir gran parte del entorno con el consecuente incremento en el tiempo de procesamiento que eso introducirá. Adicionalmente, estos sistemas se utilizan generalmente para realizar fotografías durante las rutas planeadas, que puede dar lugar a dos posibilidades: captura insuficiente o excesiva de imágenes para reconstrucción 3D.

2.1.4 Nuestro planteamiento

Habiendo analizado los diferentes sistemas de captura con imágenes terrestres en los Apartado 2.1.1 y 2.1.2 y aéreas en el Apartado 2.1.3, se buscará emular la estructura de array de cámaras mediante la grabación de una secuencia alrededor del objeto con un dron.

Esto nos puede proporcionar ciertas ventajas con respecto a otros métodos:

- **Velocidad de captura de las imágenes:** una secuencia puede grabarse en cuestión de segundos, mientras que tomar fotografías alrededor del objeto puede tomar varios minutos.
- **Número de imágenes obtenidas:** de la secuencia podremos muestrear el número de imágenes que deseemos, pudiendo muestrear más o menos imágenes según veamos el resultado de la reconstrucción.
- **Coste:** podremos simular una estructura multi-cámara con una única cámara.

También hay que señalar algunas desventajas que podemos encontrarnos:

- **Calidad de las imágenes:** la calidad de la imagen muestreada, aun en las mejores condiciones posibles, será inferior a la de una fotografía estática.
- **Emborronamiento:** algunas imágenes muestreadas puede que presenten emborronamiento debido al movimiento del dron.

2.2 Segmentación semántica

La tarea de detección de objetos en una imagen ha superado a la de clasificación de imágenes en términos de complejidad. Consiste en crear cajas delimitadoras en torno a los objetos contenidos en la imagen y clasificar cada uno de ellos. Aun así, estos modelos no suelen tener en cuenta el contexto completo de la imagen, solo clasifican parte de la información, no siendo capaces de proporcionar un entendimiento completo de la escena.

La segmentación semántica consiste en clasificar cada pixel de una imagen en una categoría correspondiente a un objeto o parte de la imagen (coche, carretera, cielo, etc.). Esta tarea es parte clave en el proceso de entendimiento de escena. Para llevarlo a cabo a día de hoy se han empezado a utilizar modelos de aprendizaje profundo (DCNN).

Para este trabajo, esta idea de clasificar cada píxel puede resultar muy útil de cara a la obtención de máscaras, ya que nos dará información bastante precisa de la posición del objeto de interés y sus límites exactos.

A continuación, vamos a presentar algunos de los modelos de segmentación semántica más populares, así como los datasets y métricas utilizados para evaluarlos.

2.2.1 Datasets y métricas

PASCAL VOC (2012) [6]. Uno de los más conocidos y usado tanto para detección de objetos y segmentación. Consta de más de 11k imágenes en sus sets de entrenamiento y validación y 10k imágenes en el de test, y 20 categorías. Para evaluar el resultado en el reto se utiliza la media de Intersección sobre Unión (mIoU). La Intersección sobre Unión es el ratio entre el área de solapamiento y el área de unión entre la ground truth y las áreas predichas. La mIoU es la media entre los objetos segmentados en todo el dataset. Las clases de PASCAL VOC 2012 pueden verse en el Anexo 2.

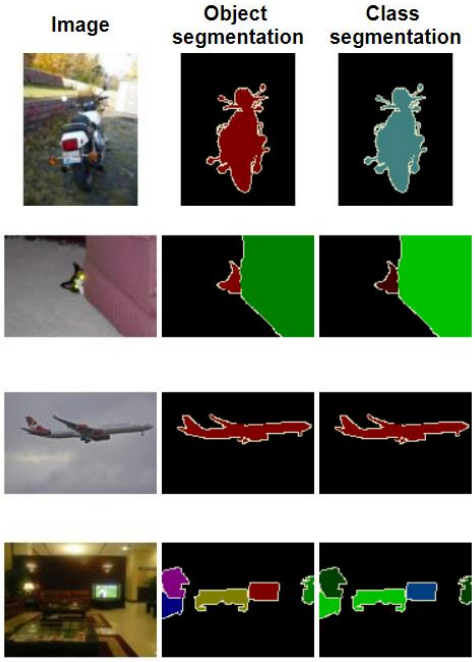


Figura 2.3. Ejemplo del dataset PASCAL VOC 2012 para segmentación de imágenes. Fuente: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

PASCAL-Context [31]. Se trata de una extensión del PASCAL VOC de 2010. Contiene 10k imágenes para entrenamiento, 10k para validación y 10k para test. Este dataset busca más especificidad introduciendo más de 400 categorías distintas. Para evaluar los resultados, se utiliza también mIoU.

Common Objects in COntext [7]. El dataset COCO consta de más de 200k imágenes con más de 500k de objetos clasificados en 80 categorías distintas. Consta de un set para entrenamiento, una validación, uno de test para investigadores y otro de test para el reto. Para su evaluación utilizado las métricas de Precisión Media (AP) y Sensibilidad Media (AR), las cuales utilizan Intersección sobre Unión (IoU).

COCO-Stuff [5]. Se trata de una versión aumentada de COCO, introduciendo anotaciones para un set de 91 clases de stuff (cosas menos reconocibles y definidas que dan contexto a

la imagen como cielo, hierba, etc.) que complementan las anotaciones de COCO para las 80 clases que ya tenía. Con esto se busca conseguir un mayor entendimiento de la escena aportando contexto a los objetos identificados en la imagen. El set de imágenes es el mismo que el de COCO 2017 con 118k imágenes para entrenamiento, 5k para validación, 5k para test para investigadores y 20k para el reto. Las clases de COCO-Stuff pueden verse en el Anexo 2.

Figura 2.4. Imágenes del dataset COCO-Stuff con anotaciones a nivel de pixel densas para *stuff* y *things*. Fuente: [5]

Cityscapes. Consiste en escenas urbanas complejas segmentadas de 50 ciudades. Está compuesta por 23.5k imágenes para entrenamiento y validación y 1.5k para test un total de 29 clases dentro de 8 super categorías: plano, humano, vehículo, construcción, objeto, naturaleza, cielo, vacío.

2.2.2 Modelos de segmentación

Fully Convolutional Network (FCN) [20]. FCN modifica arquitecturas conocidas (AlexNet [32], VGG16 [9], GoogLeNet [33]) para manejar entradas de cualquier tamaño a la vez que cambia todas las capas completamente conectadas por capas convolucionales. La red produce varios mapas de características con distintos tamaños y representaciones densas, por lo que se necesita un muestreo hacia arriba de la salida para que tenga el mismo tamaño que la entrada. Esto consiste en una convolución con un paso inferior a 1, comúnmente conocido como deconvolución al producir una salida más larga que la entrada. Adicionalmente, han añadido conexiones de salto en la red para combinar representaciones de mapa de características de alto nivel con algunas más específicas y densas en la parte superior de la red. Utilizando modelos preentrenados con el dataset ImageNet [8] de 2012 ha obtenido una puntuación en mIoU del 62.2% en el reto de PASCAL VOC 2012. En su versión Faster R-CNN ha conseguido una puntuación en mIoU del 78.8%.

ParseNet [21]. Se trata de una red convolucional de extremo a extremo que predice valores para todos los píxeles al mismo tiempo y evita tomar como entrada regiones para mantener la información global. Para ello utiliza un módulo que toma mapas de características como entrada. El primer paso utiliza un modelo para generar mapas de características que se reducen a un único vector de características global con una capa de agrupación. Este vector se normaliza utilizando la Norma Euclidiana L2 y no se agrupa (la salida es una versión ampliada de la entrada) para producir nuevos mapas de características con los mismos tamaños que los iniciales. El segundo paso normaliza los mapas de características iniciales utilizando de nuevo la Norma Euclidiana L2. Finalmente, se concatenan los mapas de características

generados. La normalización permite escalar los valores de mapas de características concatenados y proporciona un mejor rendimiento. Los resultados de ParseNet en el reto de PASCAL Context son de 40.4% mIoU y de 69,8% en el reto de PASCAL VOC 2012.

Redes convolucionales y deconvolucionales [22]. Consiste en un modelo de extremo a extremo formado por dos partes. La primera parte es una red convolucional con una arquitectura VGG16 [referencia]. Toma como entrada una propuesta de instancia (un cuadro delimitador generado por un modelo de detección de objetos). La propuesta se procesa y es transformada por una red convolucional para generar un vector de características. La segunda parte es una red deconvolucional que toma dicho vector como entrada y genera un mapa de probabilidades de píxeles que pertenecen a cada clase. La red deconvolucional utiliza desagrupamiento teniendo como objetivo las máximas activaciones para mantener su localización en los mapas. Finalmente utiliza la deconvolución para expandir los mapas de características manteniendo la información densa. Se ha observado que los mapas de características de deconvolución de bajo nivel son específicos en la forma, mientras que los superiores ayudan en la clasificación de la propuesta. Una vez todas las propuestas de una imagen son procesadas por toda la red, los mapas se concatenan para obtener la imagen completamente segmentada. Su puntuación en el reto PASCAL VOC 2012 ha sido de 72.5% mIoU.

Feature Pyramid Network (FPN) [23]. La arquitectura de la FPN se compone de una ruta de abajo hacia arriba, una ruta de arriba hacia abajo y conexiones laterales para unir características de baja y alta resolución. La ruta de abajo hacia a arriba toma una imagen con tamaño arbitrario como entrada. Se procesa con capas convolucionales y se muestrea por agrupación de capas. Siendo una etapa cada grupo de mapas de características con el mismo tamaño, las salidas de la última capa de cada etapa son las características utilizadas para el nivel de pirámide. La ruta de arriba hacia abajo consiste en el muestreo hacia arriba de los últimos mapas de características con desagrupamiento, mejorándolo con mapas de características de la misma etapa de la ruta de abajo hacia arriba con las conexiones laterales. Estas conexiones consisten en fundir los mapas de características de la ruta de abajo hacia arriba procesados con una convolución de 1x1 con los mapas de características de la ruta de arriba hacia abajo. Los mapas de características concatenados son entonces procesados por una convolución de 3x3 que produce la salida de la etapa. Finalmente, cada etapa de la ruta de arriba hacia abajo genera una predicción para detectar el objeto. Para la segmentación del objeto, se utilizan dos Multi-Layer Perceptrons (MLP) para generar dos máscaras de diferente tamaño sobre los objetos. La FPN basada en los frameworks DeepMask [25] y SharpMask [26] obtuvo unas puntuaciones de 48.1 % AR en el reto COCO 2016.

Pyramid Scene Parsing Network (PSPNet) [24]. En PSPNet, los patrones se extraen de la imagen de entrada utilizando un extractor de características con una estrategia de red dilatada [27]. Los mapas de características se introducen en un Módulo de Agrupamiento Piramidal para distinguir patrones a con diferentes escalas. Son agrupados con cuatro diferentes escalas cada uno correspondientes al nivel de la pirámide y procesados por una capa convolucional de 1x1 para reducir sus dimensiones. De esta manera, cada pirámide analiza subregiones de la imagen con diferente ubicación. Las salidas de los niveles de pirámides son muestreadas hacia arriba y concatenados con los mapas de características iniciales para terminar conteniendo toda la información de contexto local y global. Luego, son procesados por una capa convolucional para generar las predicciones a nivel de pixel.

DeepLabv2 [1]. Combina convolución dilatada, agrupamiento piramidal espacial y CRFs completamente conectados. Introduce en su modelo la misma convolución dilatada que podemos ver en [24]. Esta consiste en filtros que introducen ceros entre píxeles: si la tasa es igual a 2, el filtro introduce dos ceros, si es igual a 1, funciona como una convolución normal. La convolución dilata permite capturar múltiples escalas de los objetos. Cuando se utiliza sin agrupamiento máximo, aumenta la resolución de la salida sin aumentar el número de pesos. El Agrupamiento Piramidal Espacial Dilatado (ASPP) consiste en aplicar varias convoluciones dilatadas a la misma entrada con diferentes tasas para detectar patrones espaciales. Los mapas de características son entonces procesados en distintas ramas y concatenados utilizando interpolación bilineal para recuperar el tamaño inicial de la entrada. La salida de este proceso alimenta los CRF completamente conectados [28] detectando las fronteras entre las distintas características para producir la segmentación semántica. La mejor implementación de DeepLabv2 utilizando ResNet-101 [2] ha llegado a conseguir una puntuación de 79.7% mIoU en el reto de PASCAL VOC 2012. Para este trabajo se va a utilizar una reimplementación en PyTorch de DeepLabv2 con ResNet-101 entrenada con PASCAL VOC 2012 y COCO-Stuff 10k y 164k.

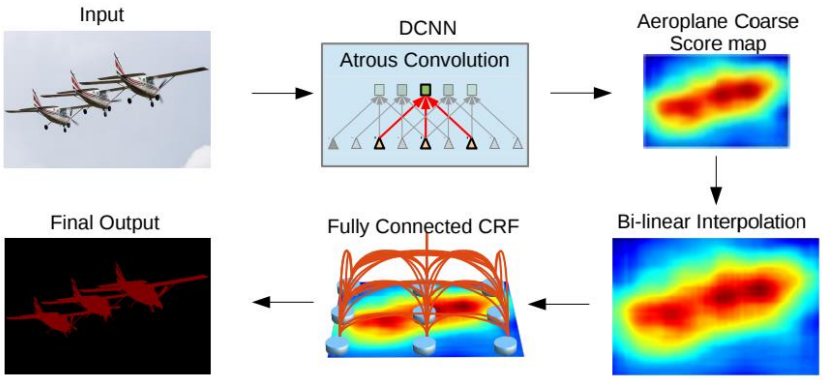


Figura 2.5. Esquema de DeepLabv2. Fuente: [1]

Modelo	PASCAL VOC 2012 (mIoU)
FCN	62.2
ParseNet	69.8
Conv & Deconv	72.5
PSPNet	85.4
DeepLabv2	79.7

Figura 2.6. Puntuaciones de los distintos modelos presentados en el reto PASCAL VOC 2012.

3 Diseño y desarrollo

3.1 Proceso completo de reconstrucción 3D a partir de video

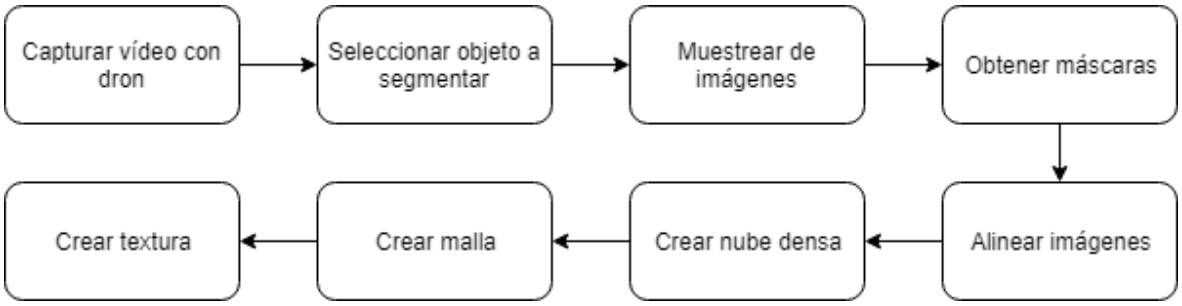


Figura 3.1. Proceso completo de reconstrucción 3D a partir de secuencia grabada con dron.

3.2 Captura de video con dron

Para la grabación de las secuencias de vídeo se ha utilizado el dron de consumo DJI Mavic Air.

Las secuencias se han grabado utilizado el modo de vuelo *Quickshot Rotation*, incorporado en el software del DJI Mavic Air, el cual realiza una circunferencia alrededor del objeto que seleccionemos manteniéndolo centrado en la imagen. Este modo tiene la limitación de necesitar una altura mínima, por lo que solo se han podido grabar objetos relativamente grandes, como personas y coches, para poder llevar a cabo las pruebas de reconstrucción.

El formato de grabación será MP4 (H.264/MPEG-4 AVC) y la resolución 4K. En el Anexo 1 pueden verse las características de la cámara del DJI Mavic Air.

3.3 Muestreo de imágenes y obtención de máscaras

3.3.1 Selección de objeto a segmentar

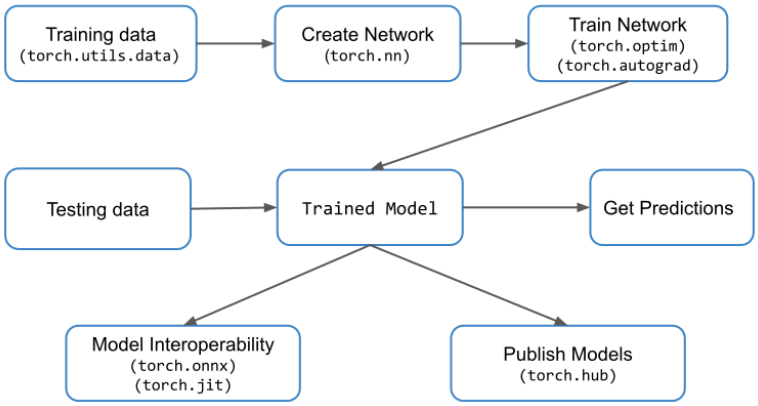


Figura 3.2. Flujo de trabajo básico de PyTorch. Fuente: https://www.learnopencv.com

Para realizar la selección del objeto del que queremos obtener máscaras, así como para realizar el propio proceso de obtención de las máscaras hemos utilizado una reimplementación de DeepLabv2 sobre la DCNN ResNet-101 en PyTorch.

PyTorch es una librería abierta basada en Python diseñada para realizar cálculos numéricos haciendo uso de programación de tensores y con soporte para utilizar GPU para cálculos. Está basado en Torch, uno de los frameworks de Deep Learning más populares a día de hoy entre la comunidad de investigación, junto con otras APIs, librerías y frameworks como Tensorflow, Caffe y Keras.

Para este trabajo se han entrenado 3 modelos distintos con los siguientes datasets: COCO-Stuff de 10k imágenes, COCO-Stuff de 164k imágenes y PASCAL VOC 12.

El proceso de segmentación realizado por DeepLabv2 puede verse en la Figura 3.3.

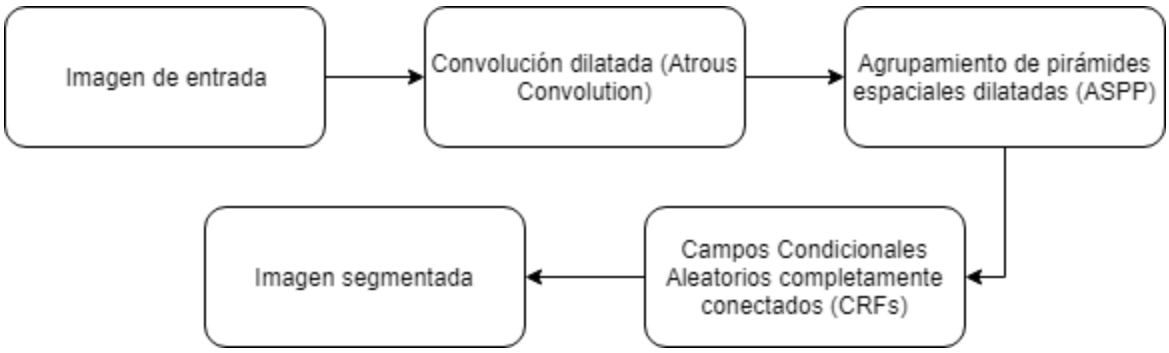


Figura 3.3. Proceso de segmentación semántica realizado por DeepLabv2. Las señales se muestrean con convolución dilatada desde 32x a 8x para extraer las características. Posteriormente se interpola para agrandar los mapas de características a la resolución original de la imagen. Finalmente, se utilizan CRF completamente conectados para refinar el resultado de la segmentación y capturar las fronteras de los objetos de mejor manera.

El uso de convolución dilatada, usada por primera vez en [10], permite aumentar la densidad de las características extraídas muestreando hacia arriba para posteriormente realizar una interpolación bilineal. En DeepLabv2 se realiza convolución dilatada para aumentar la densidad de los mapas de características calculados, seguido de una interpolación bilineal rápida con factor de 8 para recuperar los mapas de características a la resolución original de la imagen.

Posteriormente se utiliza ASPP, que utiliza múltiples capas paralelas de convolución dilatada con distintas tasas de muestreo. Las características extraídas para cada tasa de muestreo se procesan posteriormente en diferentes ramas y se funden para generar el resultado final.

Finalmente, se realiza una predicción de la estructura con CRFs completamente conectados para recuperación precisa de fronteras. Con esto, se analizan píxeles vecinos para forzar que píxeles en posiciones cercanas con colores similares lleven una misma etiqueta, consiguiendo en cada iteración ajustar más el resultado a las fronteras del objeto. DeepLabv2 realiza 10 iteraciones de posprocesado CRF.

Para realizar la selección del objeto a segmentar, se representan todas las clases detectadas, como podemos ver en la Figura 3.4, y seleccionamos la etiqueta del objeto del cual queremos obtener las máscaras.

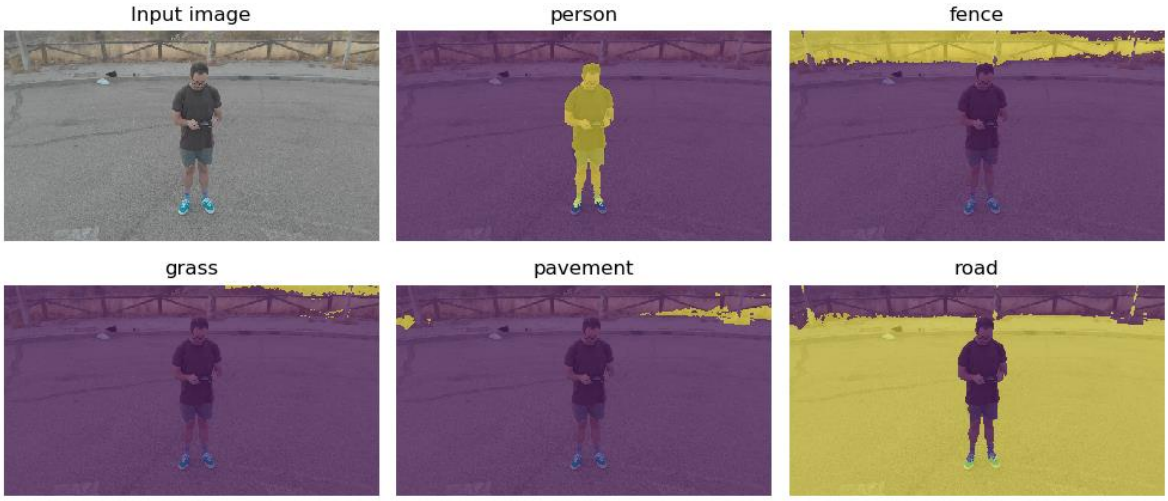


Figura 3.4. Representación de las distintas clases localizadas en la imagen analizada con el modelo de DeepLabv2 entrenado con COCO-Stuff 164k.

Podemos observar en la Figura 3.4 que se han detectado 5 clases distintas en la imagen, las cuales son:

Etiqueta	Clase
0	persona
112	valla
123	césped
139	pavimento
148	carretera

Tabla 3.1. Clases localizadas en la imagen de entrada representada en la Figura 3.2 y sus respectivas etiquetas.

3.3.2 Muestreo de imágenes

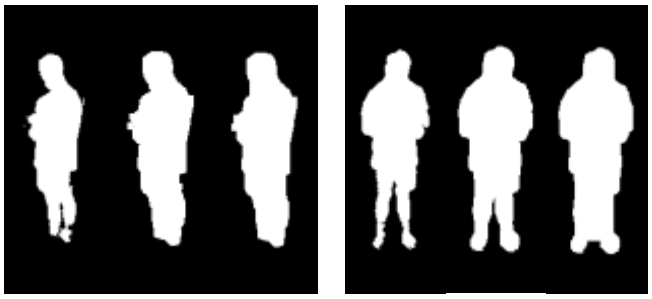
Una vez seleccionado el objeto del cual queremos obtener las máscaras, se muestrean las imágenes de la secuencia de vídeo.

El número de imágenes a muestrear será 72 ya que es el número de imágenes que mejor resultado nos ha dado para la reconstrucción, como podremos observar en el Apartado 4.4.

3.3.3 Obtención de máscaras

Para el proceso de obtención de máscaras se sigue el mismo esquema explicado en el Apartado 3.3 para la selección de objeto a segmentar, al que además se le ha añadido un procesamiento posterior a la máscara consistente en dos operaciones:

1. Dilatación con kernel 5x5, 10 iteraciones.
2. Cierre con kernel 10x10, 10 iteraciones.



(a) (b)

Figura 3.5. Proceso de dilatación y erosión sobre la máscara 1 (a) y la máscara 2 (b). La primera imagen corresponde a la máscara post-CRF, la segunda a la máscara tras dilatación y la tercera a la máscara post-cierre.

El modelo utilizado para segmentación semántica será el entrenado con PASCAL VOC 12, que nos ha proporcionado los resultados más consistentes para las secuencias utilizadas y con mayor velocidad de computación de entre los tres entrenados, como se podrá ver en el Apartado 4.2.

3.4 Reconstrucción 3D

Una vez tenemos el set de imágenes y las máscaras correspondientes, comienza el proceso de reconstrucción 3D, para el cual hemos decidido utilizar Agisoft PhotoScan. La decisión de utilizar este software viene motivada por varios factores:

1) Permite realizar todo el flujo de trabajo de reconstrucción 3D desde el alineamiento de imágenes hasta la generación de la textura de manera autónoma, sin necesidad de recurrir a otro software para complementarlo.
2) Es uno de los softwares más utilizados y que ofrece mejores resultados [13] [14] [15].
3) Permite programar el flujo de trabajo con su API de Python, dándonos la posibilidad de integrarlo en el proceso automatizado.
4) Permite importar máscaras para limitar la región de reconstrucción.

3.4.1 Alineamiento de puntos

En este paso, Agisoft PhotoScan realiza dos tareas fundamentales: parear las características extraídas a lo largo de todas las imágenes del proyecto y realizar una estimación posición y orientación de las cámaras.

Para la detección y pareado de puntos, Agisoft PhotoScan utiliza un enfoque similar al de SIFT con diferentes algoritmos que les permiten obtener mejor calidad de alineamiento. En este proceso, PhotoScan detecta puntos en las imágenes fuentes que sean estables ante variaciones de puntos de vista e iluminación y genera un descriptor para cada punto basado en su vecindario. Estos descriptores se utilizarán posteriormente para detectar correspondencias entre las imágenes.

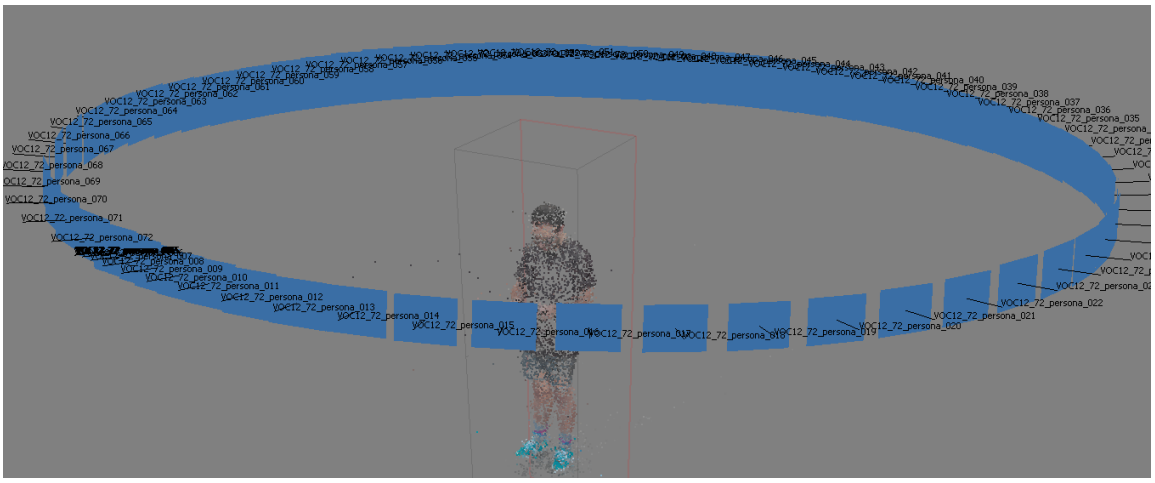


Figura 3.6. Sparse cloud, o nube escasa/poco densa de puntos junto a las posiciones de cámara estimadas. Se puede ver alrededor del objeto a reconstruir la caja delimitadora.

En este punto podemos distinguir entre dos tipos de puntos obtenidos:

- Key points o puntos clave: puntos de interés detectados en cada imagen individual.
- Tie points o puntos de unión: puntos clave compartidos en distintas imágenes.

Para el posicionamiento y orientación de las cámaras, PhotoScan utiliza algoritmos propios para hacer una estimación aproximada de la posición de las cámaras que luego refina con un ajustamiento bundle similar al de otros sistemas SfM como Bundler.

El resultado del alineamiento de puntos y posicionamiento de cameras se puede observar en la Figura 3.6.

En esa misma Figura 3.6 podemos ver la caja delimitadora, que se ha ajustado automáticamente alrededor del objeto de interés, resultado del uso de máscaras. De no haber utilizado máscaras, habría que ajustar la caja delimitadora manualmente para poder luego limitar el área de procesamiento de las siguientes operaciones. Además, el uso de máscaras ha repercutido en el tiempo de procesamiento del alineamiento de imágenes al no haber buscado puntos clave y de unión fuera del área delimitada por las máscaras. El no limitar esa área de procesamiento repercutirá en mayores tiempos de procesamiento, así como la posible aparición de texturas en el entorno que no nos interese reconstruir.

Para el alineamiento de imágenes se ha utilizado el parámetro de precisión de alineamiento High o Alto ya que utiliza la imagen original para el análisis. El valor de Precisión determinará sobre qué variante de la imagen original se realizarán las operaciones de extracción de características y pareo de puntos. Podemos ver las variantes en la Tabla 3.2.

Precisión	Imagen utilizada
Muy Alta	Imagen original * 2
Alta	Imagen original
Media	Imagen original / 2
Baja	Imagen original / 4
Muy baja	Imagen original / 8

Tabla 3.2. Imágenes utilizadas para cada distinto valor del parámetro de Precisión de alineamiento de fotos.

3.4.2 Construcción de nube densa

En este procedimiento, PhotoScan utiliza un enfoque parecido a SGM (Semi-Global Matching), realizando un cálculo de mapa de profundidad a partir de pares de imágenes que compartan un mínimo de puntos de unión para generar una nube densa de puntos. Después del cálculo de los mapas de profundidad, se realizará un filtrado para eliminar puntos que puedan ser consecuencia de imágenes ruidosas.

Calidad	Imágenes utilizadas
Ultra Alta	Imagen original
Alta	Imagen original / 2
Media	Imagen original / 4
Baja	Imagen original / 8
Muy baja	Imagen original / 16

Tabla 3.3. Imágenes utilizadas para cada distinto valor del parámetro de Calidad en la creación de nube densa de puntos.

Los parámetros utilizados en este paso han sido Ultra Alto para la calidad, por la misma razón que hemos elegido High en el apartado 3.4.1, y Moderado para el filtrado de profundidad. Podemos ver en la Tabla 3.3 la diferencia entre las imágenes utilizadas para cada calidad en el proceso de creación de nube densa, así como la diferencia entre los distintos parámetros de filtrado de profundidad en la Tabla 3.4. Se puede ver el resultado de la nube densa creada en la Figura 3.7a.

Modo	Uso
Ligero	Imágenes con detalles a reconstruir separados espacialmente del resto de la escena. Evita eliminar posibles puntos de interés.
Agresivo	Imágenes sin detalles pequeños de interés.
Moderado	Punto medio entre Ligero y Agresivo recomendado cuando no se tenga claro cuál de los otros utilizar.

Tabla 3.4. Comparación de los parámetros de filtrado de profundidad de Agisoft PhotoScan.

3.4.3 Generación de la malla

Para la generación de la malla, Agisoft PhotoScan utiliza el método de reconstrucción de superficie de Poisson apantallada [11]. Los parámetros que hemos elegido para este paso son los siguientes:

- Tipo de superficie: Arbitraria (3D)
- Fuente de datos: Nube densa
- Número de caras: Alto
- Interpolación habilitada

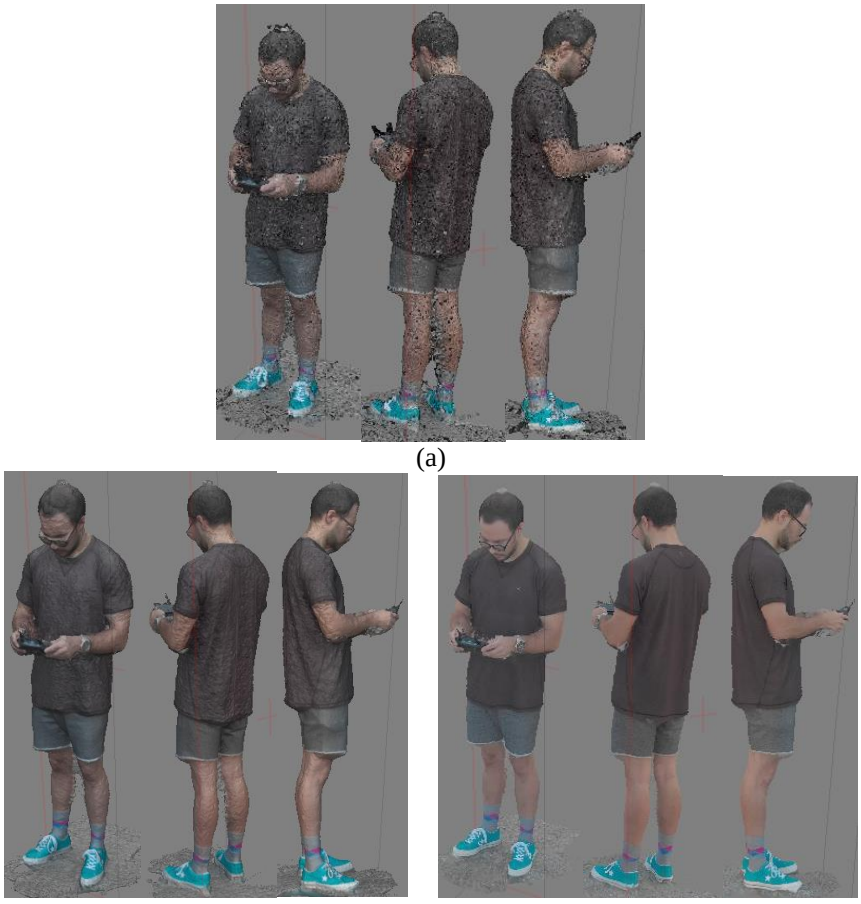
Tanto en el tipo de superficie como la fuente deben de ser elegidos esos parámetros para la reconstrucción 3D. Para el número de caras se ha elegido Alto para tener la mejor calidad posible; la relación entre los valores posibles Alto, Medio y Bajo tienen un ratio de 1/5, 1/15 y 1/45 respectivamente, utilizando los puntos calculados en la nube densa generada en el paso anterior. PhotoScan realiza una interpolación alrededor de cada punto de la nube densa para evitar que queden agujeros en la malla generada.

Podemos ver el resultado de la generación de la malla en la Figura 3.7b.

3.4.4 Creación de textura

Finalmente, se genera la textura del modelo 3D. Para este proceso, Agisoft PhotoScan parametriza la superficie y recortándola en pequeñas piezas que luego reconstruye fusionando imágenes fuente para formar el atlas de textura.

Podemos ver el resultado de este proceso en la figura 3.7c.



(a) (b) (c)

Figura 3.7. (a) Resultado de creación de nube densa de puntos en la secuencia persona con los parámetros de calidad Ultra Alto y de filtrado de profundidad Moderado. (b) Resultado de la generación de malla. (c) Modelo final con superficie texturizada.

4 Integración, pruebas y resultados

4.1 Secuencias utilizadas

Para las pruebas realizadas tanto a lo largo del proceso de desarrollo, como a la hora de evaluar los resultados, se han utilizado dos secuencias a las que nos referiremos como Persona y Coche respectivamente y podemos ver en la Figura 4.1. La secuencia Persona tiene una duración de 14 segundos y la secuencia Coche de 23 segundos.

(a) (b)

Figura 4.1. (a) Imágenes de secuencia Persona. (b) Imágenes de secuencia Coche

Para evaluar el resultado de los procesos de segmentación, se han obtenido máscaras de las imágenes de la Figura 4.1 de manera manual que podemos ver en la Figura 4.2.

(a) (b)

Figura 4.2. Máscaras obtenidas manualmente de secuencia Persona (a) y secuencia Coche (b)

4.2 Segmentación

Para decidir qué modelo de los que hemos entrenado vamos a utilizar en el proceso de reconstrucción vamos a examinar su desempeño en las secuencias de imágenes mencionadas en la sección 4.1, así como la contribución del posprocesado CRF.

Para evaluar los resultados, haremos una valoración visual que complementaremos con medidas de porcentaje de coincidencia con las máscaras obtenidas manualmente. Adicionalmente, se medirá el tiempo de procesamiento de cada modelo sobre un set de 4 imágenes.

4.2.1 Pruebas con los modelos entrenados

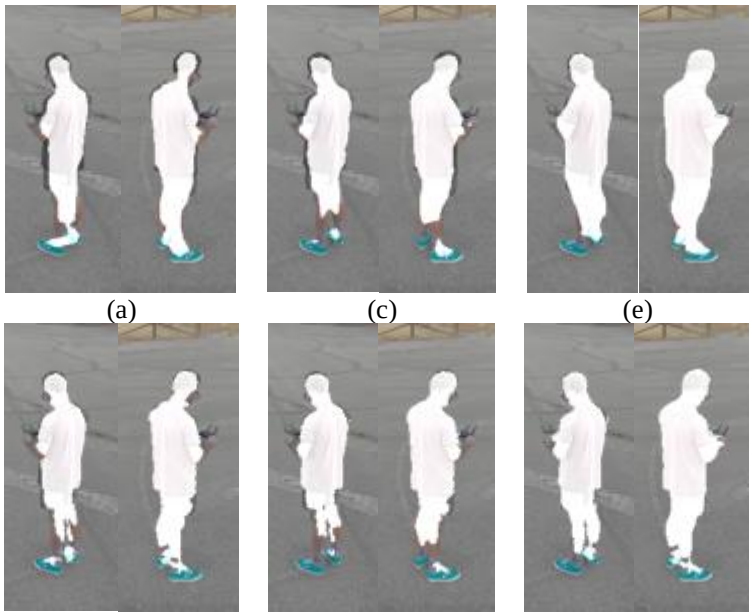


Figura 4.3. Máscaras obtenidas con los modelos de segmentación evaluados superpuestas con las imágenes originales para comparación en la secuencia Persona. (a) COCO-Stuff 10k pre-CRF y (b) post-CRF. (c) COCO-Stuff 164k pre-CRF y (d) post-CRF. (e) PASCAL VOC 12 pre-CRF y (f) post-CRF

MODELO	CRF	% máscara 1	% máscara 2	Tiempo de procesamiento
COCO-Stuff 10k	×	68.59%	89.79%	6.11s
COCO-Stuff 10k	✓	72.36%	91.36%	41.68s
COCO-Stuff 164k	×	69,13%	93,23%	6.1s
COCO-Stuff 164k	✓	72,81%	92,13%	41.56s
PASCAL VOC 12	×	84,07%	90,54%	4.94s
PASCAL VOC 12	✓	83,89%	90,79%	10.48s

Tabla 4.1. Medidas de evaluación de los modelos de segmentación utilizados sobre la secuencia Persona.

Tanto en la Figura 4.3 como en los resultados reflejados en la Tabla 4.1, podemos ver que el modelo entrenado con PASCAL VOC 12 es más consistente en cuanto a la superficie de región de interés cubierta. Asimismo, podemos ver el efecto del procesador posterior CRF a la hora de ajustar las máscaras obtenidas a los bordes de la región de interés. Respecto a los tiempos de procesamiento, PASCAL VOC 12 es el más rápido con diferencia. Esto puede ser consecuencia del hecho de manejar menos clases que COCO-Stuff.

(b) (d) (f)

Figura 4.4. Máscaras obtenidas con los modelos de segmentación evaluados superpuestas con las imágenes originales para comparación en la secuencia Coche. (a) COCO-Stuff 10k pre-CRF y (b) post-CRF. (c) COCO-Stuff 164k pre-CRF y (d) post-CRF. (e) PASCAL VOC 12 pre-CRF y (f) post-CRF

MODELO	CRF	% máscara 1	% máscara 2	Tiempo de procesamiento
COCO-Stuff 10k	×	88,40%	79,77%	6.12s
COCO-Stuff 10k	✓	87,34%	82,52%	46.62s
COCO-Stuff 164k	×	91,30%	90.58%	6.25s
COCO-Stuff 164k	✓	91.02%	91.91%	47.15s
PASCAL VOC12	×	95,21%	94,18%	5s
PASCAL VOC12	✓	93.79%	93.32%	11.44s

Tabla 4.2. Medidas de evaluación de los modelos de segmentación utilizados sobre la secuencia Coche.

Al igual que en los resultados observados para la secuencia Persona, podemos ver en la Figura 4.4. y en la Tabla 4.2 que PASCAL VOC 12 es más consistente también en el caso de la secuencia Coche.

4.2.2 Resultados de reconstrucción 3D

Figura 4.5. Resultado de reconstrucción 3D de la secuencia Persona utilizando las máscaras obtenidas con el modelo entrenado con PASCAL VOC 12.

Figura 4.6. Resultado de reconstrucción 3D de la secuencia Coche utilizando las máscaras obtenidas con el modelo entrenado con PASCAL VOC 12.

4.2.3 Conclusiones

Podemos ver en las Tablas 4.1 y 4.2 que, aunque en algunas ocasiones COCO-Stuff 164k consiga un mayor porcentaje de coincidencia con la máscara utilizada como referencia, PASCAL VOC 12 es más consistente en todas las imágenes.

Esto, unido a la diferencia en el tiempo de procesamiento de PASCAL VOC 12 frente a COCO-Stuff 10k y COCO-Stuff 164k, nos ha llevado a tomar la decisión de utilizar el modelo entrenado con PASCAL VOC 12 de ahora en adelante.

Asimismo, se ha podido observar que el posprocesado CRF, aunque en el caso de PASCAL VOC 12 no presente mejor medidas de coincidencia con las máscaras de referencia, si consigue un mejor ajuste de la máscara a los bordes del objeto de interés, el cual nos interesa mantener para eliminar la mayor parte del fondo posible.

4.3 Posprocesado de máscaras

Durante las pruebas de reconstrucción realizadas con las máscaras obtenidas inicialmente, pudimos observar que las texturas obtenidas al final del proceso de reconstrucción presentaban irregularidades con apariencia de sierra que asociamos al corte que las máscaras al no ajustarse completamente al objeto. Esto nos ha llevado a procesar las máscaras antes de ser utilizadas en la reconstrucción 3D.

Figura 4.7. Ejemplo de irregularidades en la textura en la reconstrucción de la secuencia Persona.

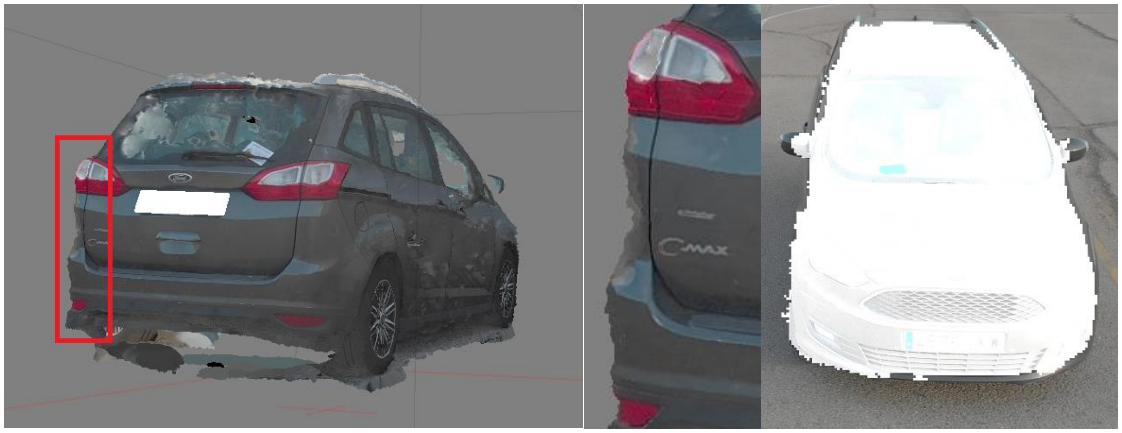


Figura 4.8. Ejemplo de irregularidades en la textura en la reconstrucción de la secuencia Coche.

4.3.1 Problemas observados

Durante el análisis que hemos realizado a las máscaras obtenidas hemos observado dos problemas a solucionar:

1. Solapamiento incompleto del objeto
2. Secciones separadas en figuras de bajo grosor (p. ej. piernas)

Estos problemas se pueden ver en la Figura 4.7, donde vemos irregularidades en los hombros, que se pueden relacionar con el solapamiento incompleto de la máscara con la región de interés que podemos observar en el resto de imágenes de la figura. Asimismo, podemos ver en la zona de las piernas como la máscara no llega hasta los pies, y en la pierna derecha incluso se pueden ver partes prácticamente desconectadas. En la Figura 4.8, podemos observar un ejemplo de estas mismas irregularidades en el lateral izquierdo del coche, que, de nuevo, podemos atribuir al solapamiento incompleto de la máscara con la región de interés, que resulta en una reconstrucción sin los bordes de la figura, especialmente en zonas ramificadas, como los retrovisores.

4.3.2 Operaciones realizadas

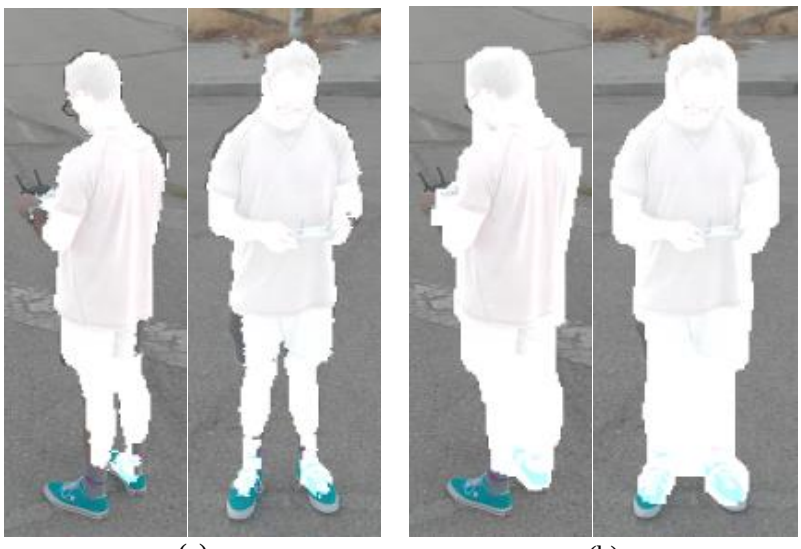
Para atacar estos problemas hemos realizado dos operaciones sobre las máscaras: dilatación y cierre. Los parámetros de cada una de las operaciones se han elegido en base a prueba y error hasta conseguir el resultado deseado.

Primero se realizan 10 iteraciones de dilatación con un kernel de 5x5, con el objetivo de agrandar la máscara y reducir la distancia entre aquellas secciones que estén separadas y acto seguido se llevan a cabo 10 iteraciones de la operación de cierre con un kernel de 10x10 con el objetivo de unir esas secciones.

(a) (b)

Figura 4.9. Proceso de dilatación y erosión sobre la máscara 1 (a) y la máscara 2 (b). La primera imagen corresponde a la máscara post-CRF, la segunda a la máscara tras dilatación y la tercera a la máscara post-cierre.

4.3.3 Resultados del posprocesado de máscaras

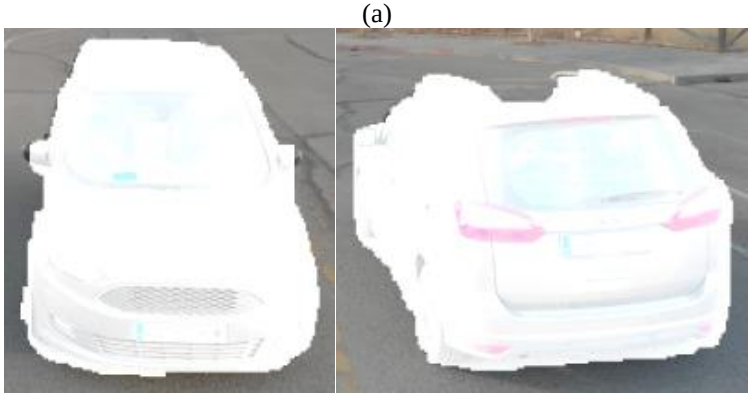


(a) (b)

Figura 4.10. Máscaras obtenidas de la secuencia Persona antes de procesar (a), y después de procesar (b)

Podemos observar en la Figura 4.10 que hemos conseguido mejorar el solapamiento de la máscara sobre la región de interés, a costa de cubrir gran parte de superficie fuera de la región de interés. Al conseguir cubrir los bordes casi en su totalidad, esperemos que se refleje en la mejora de las irregularidades en la reconstrucción 3D. No se ha conseguido, no obstante, cubrir la zona de los pies en su totalidad.

(a)



(b)

Figura 4.11. Máscaras obtenidas de la secuencia Coche antes de procesar (a), y después de procesar (b)

Al igual que en la Figura 4.10, podemos ver en la Figura 4.11 como se ha conseguido mejorar el solapamiento, cubriendo gran parte de los bordes. Se puede ver que se ha mejorado el solapamiento en la zona del techo. Los retrovisores, sin embargo, siguen siendo un problema.

CRF	Posprocesado	Máscara 1	Máscara 2
×	×	84,07%	90,54%
×	✓	90.57%	97,79%
✓	×	83,89%	90,79%
✓	✓	91,85%	97,34%

Tabla 4.3. Medidas de porcentaje de coincidencia de máscaras obtenidas para Persona con las de referencia de la Figura 4.2.

CRF	Posprocesado	Máscara 1	Máscara 2
×	×	95,21%	94,18%
×	✓	98.43%	98.34%
✓	×	93.79%	93.32%
✓	✓	97,47%	98,39%

Tabla 4.4. Medidas de porcentaje de coincidencia de máscaras obtenidas para Coche con las de referencia de la Figura 4.2.

(a) (b)

Figura 4.12. Comparación de la reconstrucción 3D de la secuencia Persona pre y post procesado de las máscaras. (a) Reconstrucción con máscaras sin procesar. (b) Reconstrucción con máscaras procesadas. Podemos observar en las mangas como en (b) se ha conseguido una ligera mejora en la suavidad de la textura de los hombros con respecto a las irregularidades observadas en (a).

(a) (b)

Figura 4.13. Comparación de la reconstrucción 3D de la secuencia Coche pre y post procesado de las máscaras. (a) Reconstrucción con máscaras sin procesar. (b) Reconstrucción con máscaras procesadas. Podemos observar en el lateral izquierdo del coche como en (b), en este caso, se ha conseguido una notable mejora en la suavidad de la textura con respecto a las irregularidades observadas en (a).

4.3.4 Conclusiones

Como hemos podido ver en las Figuras 4.10b y 4.11b, se ha conseguido el objetivo de evitar un solapamiento incompleto de los objetos segmentados casi en su totalidad, así como unir secciones que quedaron separadas en la máscara original. Con esto hemos reducido considerablemente el número de porciones del objeto que se han perdido en el proceso de reconstrucción y menos irregularidades en las texturas reconstruidas, como puede apreciarse en las Figuras 4.12b y 4.13b.

No obstante, en los casos en los que el objeto reconocido tenga “ramas” como pueden ser las piernas y brazos que no hayan sido reconocidos completamente, es difícil llegar a extender las máscaras y cubrirlos con las operaciones realizadas.

4.4 Número de imágenes muestreadas

El número de imágenes a muestrear es el factor más importante de todo el proceso, ya que cuanto mayor sea el número, más tiempo de procesamiento van a tomar los procesos realizados sobre ellas. Necesitamos encontrar un número de imágenes suficiente para que PhotoScan sea capaz de encontrar puntos de interés y alinearlos correctamente, pero asegurándonos que sea el menor número de imágenes posible para generar consistentemente modelos 3D de calidad.

4.4.1 Método de valoración

Para valorar qué número de imágenes da mejor resultado vamos a tener en cuenta varios factores: (1) número de puntos de unión, (2) número de puntos de nube densa, (3) tiempo de procesamiento. Además, se hará una valoración visual para valorar el resultado de alineamiento de puntos. En algunos casos, aunque aparentemente las cámaras parezcan bien posicionadas, los puntos no se habrán alineado correctamente.

Empezaremos en 24 imágenes y haremos pruebas hasta 120, utilizando múltiplos de 12. En aquellos que den buen resultado de reconstrucción 3D, valoraremos la diferencia entre puntos detectados, tiempo de procesamiento y resultado visual de la reconstrucción 3D.

Para la reconstrucción 3D utilizaremos los parámetros de reconstrucción siguientes:

- Alineamiento de fotos: Alta
- Calidad de construcción de nube densa: Media
- Filtrado de profundidad: Ligero

4.4.2 Resultados de reconstrucción 3D

Núm. imágenes	Posicionamiento cámaras correcto	Puntos de unión	Puntos nube densa	Tiempo de reconstrucción
24	Si*	1.648	83.308	1m 4s
48	Si*	4.901	144.120	1m 59s
60	Si	6.592	168.838	2m 37s
72	Si	7.747	144.773	2m 57s
96	No	4.788	35.377	3m 5s
120	No	6.693	87.553	4m 16s

Tabla 4.5. Datos de reconstrucción 3D para distinto número de imágenes para el objeto Persona 3.

Núm. imágenes	Posicionamiento cámaras correcto	Puntos de unión	Puntos nube densa	Tiempo de reconstrucción
24	No	845	103.390	1m 7s
48	No	1.820	119.524	2m 4s
60	Si*	7.722	272.697	3m 10s
72	Si	10.291	3.239.935	4m
96	No	13.174	300.064	5m 23s
120	No	5.752	58.352	4m 10s

Tabla 4.6. Datos de reconstrucción 3D para distinto número de imágenes para el objeto Coche.

*Posicionamiento de cámaras aparentemente correcto, pero puntos mal alineados.

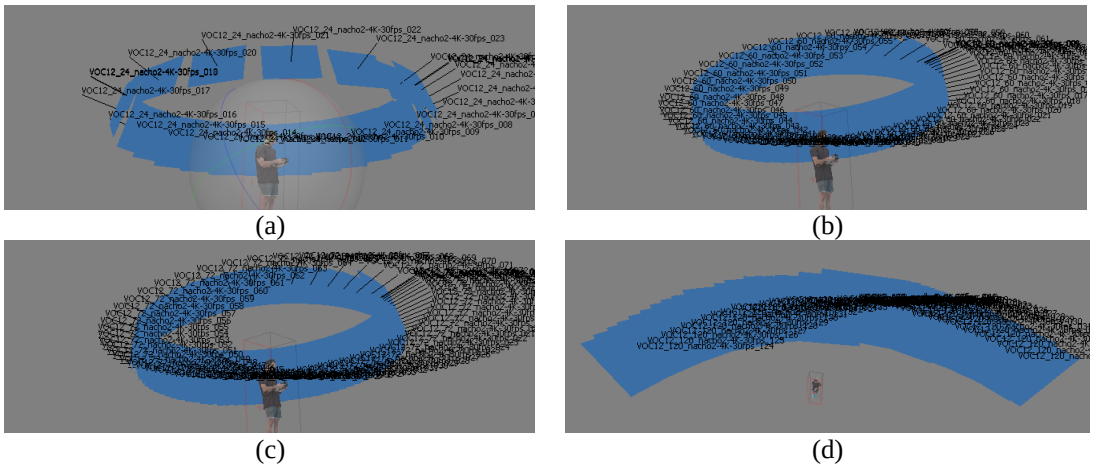


Figura 4.14. Alineamiento de cámaras de la secuencia Persona para distintos números de imágenes. (a) 24 imágenes. (b) 60 imágenes. (c) 72 imágenes. (d) 120 imágenes.

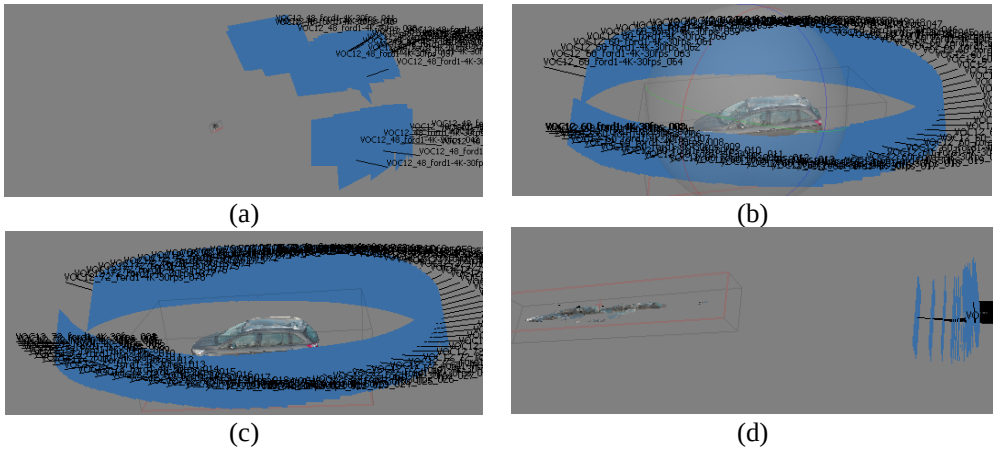


Figura 4.15. Alineamiento de cámaras de la secuencia Coche para distintos números de imágenes. (a) 48 imágenes. (b) 60 imágenes. (c) 72 imágenes. (d) 120 imágenes.

4.4.3 Conclusiones

Hemos podido ver, especialmente en los resultados de la secuencia Coche visto en la Figura 4.15, que el mejor valor es un número intermedio. Aunque aparentemente en los casos de 60 y 72 imágenes de la secuencia Coche, el posicionamiento de las cámaras sea correcto, como podemos observar en las Figuras 4.15b y 4.15c, en el caso de 60 imágenes no ha habido alineamiento correcto de puntos en algunas zonas de la reconstrucción. Por tanto, por la consistencia observada en ambas secuencias, 72 imágenes ha sido el número elegido.

4.5 Proceso completo de reconstrucción 3D desde el vídeo

Para terminar, vamos a hacer una comparación entre una reconstrucción de 3D a partir del vídeo sin realizar ningún proceso y el resultado de la reconstrucción a partir del mismo vídeo, pero siendo procesada por el programa que hemos diseñado, de manera que podamos ver el efecto de las operaciones que hemos integrado.

En estos procesos se han utilizado los siguientes parámetros de construcción:

- Precisión de alineamiento: Alta
- Calidad de nube densa: Ultra Alta

- Filtrado de profundidad: Moderado

4.5.1 Resultados de reconstrucción 3D

Para evaluar los resultados finales de reconstrucción 3D, vamos a hacer una comparación visual, así como analizar medidas de procesamiento para ver las diferencias, ventajas y desventajas de tres casos:

1) Reconstrucción 3D automatizada sin uso de máscaras.
2) Reconstrucción 3D manual sin uso de máscaras y ajuste de caja delimitadora manual.
3) Reconstrucción 3D automatizada con uso de máscaras.

(a) (b) (c)

Figura 4.16. Resultados de reconstrucción 3D de la secuencia Persona. (a) Reconstrucción automática sin uso de máscaras. (b) Reconstrucción manual sin uso de máscaras y ajuste de la caja delimitadora antes de crear la nube densa. (c) Reconstrucción automática con uso de máscaras.

(a) (b)

(c)

Figura 4.17. Resultados de reconstrucción 3D de la secuencia Coche. (a) Reconstrucción automática sin uso de máscaras. (b) Reconstrucción manual sin uso de máscaras y ajuste de la caja delimitadora antes de crear la nube densa. (c) Reconstrucción automática con uso de máscaras.

Objeto	Uso de máscaras	Caja delimitadora	Tiempo de reconstrucción 3D	Tiempo total	Puntos de unión	Puntos nube densa
Persona	×	×	5h 6m 41s	5h 14m 38s	330.155	42.503k
Persona	×	✓	42m 24s	-	330.155	2.934k
Persona	✓	✓	13m 13s	18m 43s	7.700	1.954k
Coche	×	×	9h 43m 33s	9h 49m 52s	231.778	42.074k
Coche	×	✓	52m 29s	-	231.778	7.784k
Coche	✓	✓	20m 16s	27m 22s	10.271	3.452k

Tabla 4.7. Medidas de procesamiento de reconstrucción 3D de los distintos casos mostrados en las Figuras 4.16 y 4.17.

4.5.2 Evaluación del resultado final

Podemos ver en las Figuras 4.16a y 4.17a que la reconstrucción del objeto de interés no se ha completado correctamente. Sin embargo, podemos ver en la Tabla 4.7 que tienen más puntos de unión que las reconstrucciones en las que se han usado máscaras, lo cual es debido a que esos puntos no pertenecen necesariamente al objeto de interés. El resultado incompleto de la reconstrucción es debido a que Agisoft PhotoScan ha discriminado puntos de unión encontrados en el objeto de interés en favor de puntos de unión encontrados en el fondo a la hora de generar la nube densa de puntos.

En las Figuras 4.16.b, 4.16c, 4.17b y 4.17c podemos observar que los dos modelos obtenidos son muy similares. Haciendo una evaluación visual, se puede apreciar que el modelo generado sin máscaras presenta irregularidades en la textura como las observadas en el Apartado 4.3, así como artefactos no deseados en la zona de la cabeza y ligeramente en otros puntos del modelo. Esto último se debe al ruido que puede introducir el fondo en entornos no controlados y que, en este caso, encuentra puntos de unión en zonas adyacentes a los bordes del objeto de interés. En la Tabla 4.7 podemos observar también que el tiempo de procesamiento es significativamente superior, debido sobre todo a que, aun habiendo limitado la zona de reconstrucción, el suelo contiene muchos puntos de unión de los cuales también se reconstruirán puntos para la nube densa.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

5.1.1 Trabajo realizado

Este trabajo tenía como objetivo principal desarrollar un sistema de reconstrucción 3D a partir de imágenes captadas por drones. De este objetivo principal se pudieron establecer varios objetivos intermedios:

- Optimización del tiempo de captura de imágenes
- Segmentación del objeto a reconstruir
- Desarrollo de proceso automatizado de reconstrucción 3D

Hemos reducido el tiempo de captura de imágenes utilizando un dron para grabar un video alrededor del objeto de interés. Esto nos ha permitido tener acceso a un gran número de imágenes con gran solapamiento entre sí en cuestión de segundos, como se ha mencionado en el apartado 4.1.

Para desarrollar el proceso completo de reconstrucción desde el vídeo, se decidió integrar el software de reconstrucción 3D Agisoft PhotoScan, el cual incorpora una API de Python para desarrollar distintos procesos de reconstrucción a gusto del usuario. Para decidir qué condiciones debía cumplir el proceso completo se llevó a cabo un proceso de análisis de PhotoScan y el efecto de los distintos parámetros de reconstrucción.

Para reducir el tiempo de procesamiento, así como mejorar el resultado final del objeto reconstruido y minimizar el procesamiento posterior se decidió segmentar el objeto de interés del fondo. Para ello se decidió utilizar algoritmos de segmentación semántica a partir de modelos entrenados con diversos datasets. En este trabajo utilizamos una reimplementación de DeepLabv2 en PyTorch con soporte para CRFs. A pesar del buen resultado que daban estos procesos, se decidió añadir un procesamiento posterior a cada máscara para cubrir en la mayor medida posible el objeto y obtener mejores resultados en PhotoScan.

Finalmente, se automatizó el proceso de reconstrucción 3D a partir del video en un script de Python, el cual, a su vez, lanzaba el script de Python para controlar el flujo de trabajo de reconstrucción del objeto 3D en PhotoScan. Este proceso se ha desarrollado pensando en liberar al usuario de tener que interactuar con él en pasos intermedios, seleccionando las imágenes a muestrear, los parámetros de reconstrucción y el objeto a segmentar al comienzo del proceso. El resultado obtenido al final de la ejecución será un proyecto de Agisoft PhotoScan con la reconstrucción 3D del cual se podrá editar y exportar la información que se quiera utilizar.

5.1.2 Resultados obtenidos

Respecto a los objetivos planteados al comienzo del proyecto y mencionados en el primer punto de este apartado, pudiera decirse que se han cumplido todos, aunque haya un gran espacio para la mejora:

- Se ha optimizado el tiempo de captura de imágenes utilizando vídeos captados con drones utilizando un método de grabación de video incluido en el software del dron.

- Se ha logrado obtener máscaras del objeto a construir utilizando segmentación semántica. El resultado de esta segmentación ha sido bueno para objetos grandes y definidos como han sido personas y coches. No es tan bueno para objetos con características menos reconocibles.
- Se ha desarrollado un script que automatiza todo el proceso de reconstrucción a partir del vídeo capturado.

Se ha podido ver que los resultados obtenidos pueden mejorarse dedicando más tiempo a determinados procesos que se han llevado a cabo en este trabajo con menor profundidad, como pudieran ser el proceso de captura de las imágenes o el proceso de segmentación del objeto.

5.1.3 Limitaciones observadas

Durante el desarrollo de este trabajo se han podido observar las siguientes limitaciones que han impedido obtener mejores resultados:

1) **Grabación de vídeo con dron.** El modo de vuelo utilizado limita la resolución en la reconstrucción al solo tener una vista superior. Para mejor captura se propone programar ruta del dron. Vuelo manual no se contempla ya que no aporta consistencia.
2) **Obtención de máscaras.** Como se ha observado en el apartado 4.3 aunque con el posprocesado introducido se han mejorado los resultados para las máscaras a utilizar en el proceso de reconstrucción 3D, no se ha conseguido cubrir partes del objeto no reconocidas por completo como pueden ser aquellas que tienen forma de rama, como serían brazos o piernas. Para ello se propone probar con otros métodos y modelos de segmentación.
3) **Reconocimiento de objetos.** Los datasets utilizados para entrenar las redes neuronales son muy buenos para imágenes naturales y objetos comunes bien definidos, pero tal vez interesaría buscar algún dataset destinado a reconocer objetos centrados en la imagen, ya que nuestro objetivo será siempre tener el objeto centrado a la hora de capturarlo.

(a) (b)

Figura 5.1. Ejemplo del efecto de solo tener el punto de vista superior y como afecta a la reconstrucción. (a) Vista superior de la secuencia reconstruida. (b) Vista inferior de la misma zona donde se puede ver el efecto de no tener imágenes captadas desde otras alturas.

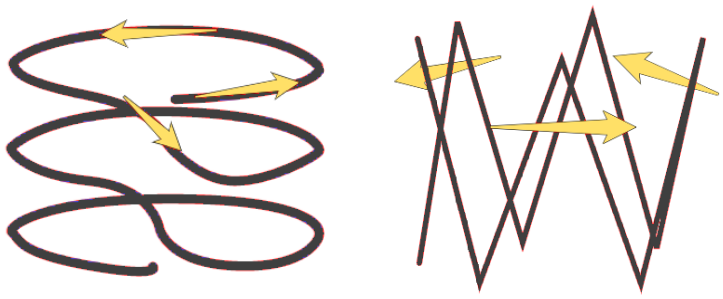
5.2 Trabajo futuro

5.2.1 Optimización de la captura mediante programación de ruta de vuelo de dron

En este proyecto nos hemos centrado principalmente en la parte de automatización del proceso de reconstrucción 3D, así como en mejorar los resultados del proceso automatizado eliminando el fondo mediante el uso de segmentación semántica. Por ello, la ruta que hemos realizado con el dron durante las grabaciones ha sido muy básica, consistente en una única vuelta alrededor del objeto a una altura constante.

No obstante, una manera de mejorar los resultados de la reconstrucción 3D es realizando vuelos más complejos que consigan más posiciones de la cámara y solapamiento vertical aparte del solapamiento horizontal que hemos usado nosotros. Esto se puede conseguir programando la ruta del dron, lo que conseguiría no solo mejorar los resultados, sino también mejorar en sí el proceso de captura, la velocidad y automatizarlo.

Algunos ejemplos de la ruta que podría programarse se pueden ver en la Figura 5.1:



(a) (b)

Figura 5.2. Ejemplo de rutas programables para la captura de secuencias para reconstrucción 3D. Ruta en espiral (a). Ruta en zig-zag (b).

Estas rutas de vuelo representadas en la figura 5.1 pueden ayudar a superar las limitaciones mencionadas en la sección 5.1.3 con respecto a la grabación de la secuencia, aportando no solo resolución horizontal para la reconstrucción 3D, sino también resolución vertical.

Asimismo, sería de especial interés crear un dataset para valoración más objetiva de trabajos relacionados compuesto de:

- Secuencias de video de objetos capturadas con las rutas de vuelo programadas.
- Fotografías de alta calidad de los mismos objetos para generar modelos 3D de máxima calidad para utilizar como referencias
- Sets de imágenes muestreadas de las secuencias con máscaras correspondientes obtenidas a mano.

5.2.2 Refinamiento y optimización del proceso de reconstrucción 3D

Este trabajo ha sido un primer acercamiento a buscar una solución de bajo coste para reconstrucción 3D tratando de maximizar la calidad del resultado y optimizando la velocidad de reconstrucción incorporando distintas tecnologías disponibles en la actualidad.

Por lo tanto, puede mejorarse notablemente aportando otros enfoques para cualquiera de las distintas partes del proceso, desde la captura del video (como proponemos en el Apartado 5.2.1), procesamiento de las imágenes utilizando distintos modelos de segmentación (como proponemos en el apartado 5.1.3) así como realizar modificaciones sobre las máscaras obtenidas con dichos modelos; hasta el software de reconstrucción 3D utilizado, donde se propone desarrollar alternativas con software de código libre.

Adicionalmente, se propone evaluar la eficacia tanto de este trabajo como de los distintos trabajos relacionados que lo sucedan utilizando el dataset cuya creación se ha propuesto en el Apartado 5.2.1, de manera que puedan medirse correcta y objetivamente las distintas mejoras que se realicen sobre este trabajo.

Referencias

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *IEEE TPAMI*, 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.

[3] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in NIPS, 2011.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in ECCV, 2014.

[5] H. Caesar, J. Uijlings, V. Ferrari. "COCO-Stuff: Thing and Stuff Classes in Context." In *CVPR*, 2018.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. "The PASCAL Visual Object Classes (VOC) Challenge." *IJCV*, 2010.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Zitnick. "Microsoft COCO: Common objects in context." In ECCV, 2014.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. "ImageNet large scale visual recognition challenge." IJCV, 2015.

[9] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In ICLR, 2015.

[10] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in Proc. Wavelets: Time-Frequency Methods Phase Space, 1989, pp. 289–297.

[11] M. Kazhdan and H. Hoppe. 2013. Screened poisson surface reconstruction. ACM Trans. Graph. 32, 3, Article 29 (July 2013), 13 pages. DOI: https://doi.org/10.1145/2487228.2487237

[12] Nex, F & Remondino, F. (2014). "UAV for 3D mapping applications: A review. Applied Geomatics." 6. 10.1007/s12518-013-0120-x.

[13] Brocks, S & Bareth, G. (2016). "EVALUATING DENSE 3D RECONSTRUCTION SOFTWARE PACKAGES FOR OBLIQUE MONITORING OF CROP CANOPY SURFACE." ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLI-B5. 785-789. 10.5194/isprsarchives-XLI-B5-785-2016.

[14] Alouache, A. & Yao, X. & Wu, Q. (2017). “Creating Textured 3D Models from Image Collections using Open Source Software.” International Journal of Computer Applications. 163. 14-19. 10.5120/ijca2017913734.

[15] Murtiyoso, A.; Grussenmeyer, P.; Börlin, N.; Vandermeerschen, J.; Freville, T. “Open Source and Independent Methods for Bundle Adjustment Assessment in Close-Range UAV Photogrammetry.” *Drones* 2018, *2*, 3.

[16] Mahiddine, A. & Seinturier, J. & Daniela, P. & Boulaassal, H. & Boï, J. & Merad, D. & Drap, P. (2013). “Validating photogrammetric orientations steps by the use of relevant theoretical models. Implementation in the ‘Arpenteur’ framework.” ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XL-5/W2. 10.5194/isprsarchives-XL-5-W2-409-2013.

[17] Pix4Dmodel: 3D model and measure the world with images | Pix4D. https://www.pix4d.com/product/pix4dmodel

[18] Drone & UAV Mapping Platform | DroneDeploy. https://www.dronedeploy.com/

[19] Drone Mapping Software – OpenDroneMap. https://www.opendronemap.org/

[20] Shelhamer, E., J. Long, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation.” IEEE Transactions on Pattern Analysis and Machine Intelligence 39.4 (2017): 640–651. Crossref. Web.

[21] Liu, W., Rabinovich, A., & Berg, A.C. (2015). “ParseNet: Looking Wider to See Better.” *ArXiv, abs/1506.04579.*

[22] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. “Learning Deconvolution Network for Semantic Segmentation.” 2015 IEEE International Conference on Computer Vision (ICCV) (2015): n. pag. Crossref. Web.

[23] Lin, Tsung-Yi et al. “Feature Pyramid Networks for Object Detection.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): n. pag. Crossref. Web.

[24] Zhao, Hengshuang et al. “Pyramid Scene Parsing Network.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): n. pag. Crossref. Web.

[25] Pinheiro, P.H., Collobert, R., & Dollár, P. (2015). Learning to Segment Object Candidates. *NIPS*.

[26] Pinheiro, P.H., Lin, T., Collobert, R., & Dollár, P. (2016). Learning to Refine Object Segments. *ECCV*.

[27] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

[28] Krähenbühl, P., & Koltun, V. (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. *NIPS*.

[29] How Do Drones Work And What Is Drone Technology | DroneZon. https://www.dronezon.com/learn-about-drones-quadcopters/what-is-drone-technology-or-how-does-drone-technology-work/

[30] Pixel Light Effects - Providing on set VFX 3D Scanning. https://pixellighteffects.com.

[31] Mottaghi, Roozbeh & Chen, Xianjie & Liu, Xiaobai & Cho, Nam-Gyu & Lee, Seong-Whan & Fidler, Sanja & Urtasun, Raquel & Yuille, Alan. (2013). "The Role of Context for Object Detection and Semantic Segmentation in the Wild." 10.13140/2.1.2577.6000.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[33] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.

Anexos

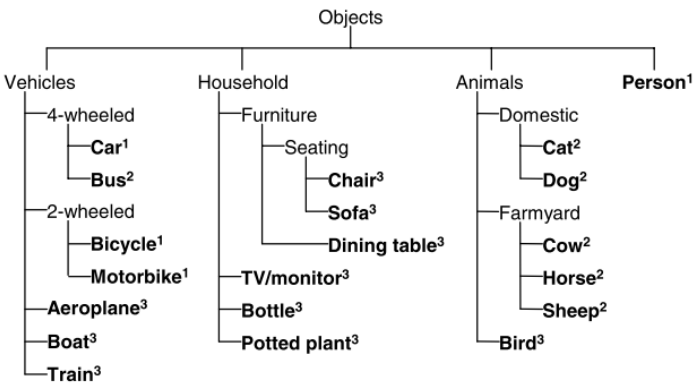
Anexo 1. Características de la cámara de Mavic Air

Sensor	1.2/3" CMOS Píxeles efectivos: 12 MP
Objetivo	FOV: 85º Formato equivalente a 35mm: 24 mm Apertura: f/2.8 Distancia de enfoque: 2 m a
Rango ISO	Vídeo: 100 - 3200 (automático) 100 - 3200 (manual) Foto: 100 - 1600 (automático) 100 - 3200 (manual)
Velocidad de obturación	Obturador electrónico: 8 - 1/8000s
Tamaño de fotografía	4:3: 4056x3040 16:9: 4056x2280
Modos de fotografía	Disparo único HDR Disparo en ráfaga: 3/5/7 fotogramas Exposición automática en horquillado (AEB), 3/5 horquillas de exposición a 0.7 EV bias Intervalo: 2/3/5/7/10/15/20/30/60 s Pano: 3x1: 42ºx78º, 2048x3712 (Ancho x Alto) 3x3: 119ºx78º, 4096x2688 (Ancho x Alto) 180º: 25ºx88º, 6144x2048 (Ancho x Alto) Esfera (3x8+1): 8192x4096 (Ancho x Alto)
Resolución de vídeo	4K Ultra HD: 3840x2160 24/25/30p 2.7K: 2720x1530 24/25/30/48/50/60p FHD: 1920x1080 24/25/30/48/50/60/120p HD: 1280x720 24/25/30/48/50/60/120p
Tasa de bits máx de almacenamiento de vídeo	100 Mbps
Sistema de archivos compatibles	FAT32
Formatos de fotografía	JPEG/DNG (RAW)
Formatos de vídeo	MP4/MOV (H.264/MPEG-4 AVC)

Anexo 2. Etiquetas de datasets

COCO-Stuff

VOC12



Glosario

GPU	Graphics Processing Unit
API	Application Programming Interface
DCG	Dynamic Computation Graph
DCNN	Deep Convolutional Neural Network
COCO	Common Objects in Context
COCO-Stuff	Common Objects in Context and Stuff
VOC	Visual Object Classes
UAV	Unmanned Aerial Vehicle
SIFT	Scale-invariant feature transform
SfM	Structure from Motion
SGM	Semi-Global Matching